International Journal of Neural Systems, Vol. 0, No. 0 (2021) 1–23

TILED SPARSE CODING IN EIGENSPACES FOR IMAGE CLASSIFICATION

JUAN E ARCO[1,4], ANDRÉS ORTIZ[2,4], JAVIER RAMÍREZ[1,4], YU-DONG ZHANG [3], JUAN M GÓRRIZ[1,4]
[1]*Department of Signal Theory, Networking and Communications, University of Granada, 18010 Spain*

[2]*Department of Communications Engineering, University of Malaga, 29010 Spain*

[3]*School of Informatics, University of Leicester, Leicester, LE1 7RH, UK*

[4]*Andalusian Research Institute in Data Science and Computational Intelligence, Spain*

E-mail: jearco@ugr.es

The automation in the diagnosis of medical images is currently a challenging task. The use of Computer Aided Diagnosis (CAD) systems can be a powerful tool for clinicians, especially in situations when hospitals are overflowed. These tools are usually based on artificial intelligence (AI), a field that has been recently revolutionized by deep learning approaches. These alternatives usually obtain a large performance based on complex solutions, leading to a high computational cost and the need of having large databases. In this work, we propose a classification framework based on sparse coding. Images are first partitioned into different tiles, and a dictionary is built after applying PCA to these tiles. The original signals are then transformed as a linear combination of the elements of the dictionary. Then, they are reconstructed by iteratively deactivating the elements associated with each component. Classification is finally performed employing as features the subsequent reconstruction errors. Performance is evaluated in a real context where distinguishing between four different pathologies: control *vs* bacterial pneumonia *vs* viral pneumonia *vs* COVID-19. Our system differentiates between pneumonia patients and controls with an accuracy of 97.74%, whereas in the 4-class context the accuracy is 86.73%. The excellent results and the pioneering use of sparse coding in this scenario evidence that our proposal can assist clinicians when their workload is high.

Keywords: Computer-aided diagnosis, medical imaging, machine learning; deep learning, sparse coding; dictionary; pneumonia; COVID-19.

1. Introduction

Nowadays, medical images are extremely relevant for the diagnosis of a high number of diseases. The information provided by these images can be analyzed solely or combined with other complementary tests. In both cases, a proper interpretation of the images is crucial for avoiding the delay in the diagnostic process, which can risk the patient's health. With the advance of artificial intelligence (AI), computer aided diagnosis (CAD) systems have been successfully used in the study of different pathologies. The idea behind these methods is the identification of a pattern in new samples based on the information acquired in previous data.[1–5] The simplest approach analyzes the information contained in group of pixels/voxels (for bidimensional and three-dimensional images, respectively) from raw data. Working in the original space makes it necessary to handle a large amount of features. Besides, the number of samples is usually much lower, originating the so-called curse of dimensionality problem.[6] To address this issue, two

main alternatives have been developed in order to reduce the dimensionality of the feature space. The first one consists on identifying the most discriminative features, discarding the rest for the subsequent analysis.[7–9] The second relies on computing a new set of features of a lower dimension than the original space.[10–12]

The suitability of these intelligent systems depends on the context. The ultimate aim of CAD frameworks is the classification of a specific pathology with the largest performance as possible. However, there are other important factors to take into account when designing a classification system, such as the interpretability of the model. For example, a CAD framework for the prediction of Alzheimer's disease (AD) must differentiate between brain damage due to aging from the atrophy caused by a cognitive impairment. Besides, it is extremely useful to identify the brain regions that are first affected by this neurological disorder, allowing clinicians to increase their knowledge about this disease. This duality between performance and spatial information has been successfully met in previous studies.[13,14] Following the example of AD, Ref.[15] presents a robust ensemble scheme that combines images of different modalities for an early diagnosis of this disease. Specifically, they determined the role of each brain region in the first stages of AD according to the anatomical information derived from an atlas. Other recent works propose the use of fractals for generating informative features for the classification of Parkinson's disease (PD).[16] This kind of methodologies, based on machine learning, has also been applied in the diagnosis of different pathologies such as autism spectrum disorder (ASD),[17–19] epilepsy[20–22] or cancer.[23–25]

The emergence of deep learning and the advances in computation have revolutionized the biomedical image processing.[26–33] Convolutional neural networks (CNN) have particularly undergone a real step forward in image classification.[34–42] CNNs are biologically-inspired models that compute image features at different abstraction levels. They are based on convolution, a mathematical operation which is subsequently applied to the response of the previous layer in a neural network.[43] These studies provide good results compared to other machine learning algorithms, and their use in medical images is widely established.[1,44–53] The main drawback of these architectures is the high complexity that they usually entail. This causes that networks tend to be overfitted because of the limitations of the training algorithms, leading to a drop in performance.[54] Moreover, the generalization ability of these alternatives highly depends on the size of the dataset from which the model is trained. When the number of samples is high, methods based on deep learning usually learn the main features that characterize the different classes to distinguish from. However, it is much more difficult to learn the relationship between samples and labels when data is limited. Thus, the optimum performance of deep learning alternatives would be reached in scenarios where a vast number of images are available.

Despite the rising of public datasets, the requirements needed for the correct use of deep learning approaches can not always be met. In this work, we provide an alternative framework for image classification based on sparse coding.[55] First, images are partitioned into a number of squared tiles, and a dictionary is built after applying PCA to a matrix where all the tiles are stored. These tiles are coded from their original space to the new one created as a linear combination of the elements of the dictionary. After that, tiles are reconstructed in a process where atoms associated with different components are iteratively deactivated. The differences in the reconstruction errors in each iteration produce patterns that are extremely informative. In fact, we hypothesize that these patterns are different for the pathologies studied in this work. Finally, errors are entered into a classifier, deciding the label of each individual sample. Performance is evaluated in a real scenario where trying to identify the presence of pneumonia in chest X-ray (CXR) images. Specifically, our method is applied in four contexts of incremental difficulty: from a control *vs* pneumonia patients to a multiclass context where differentiating between four pathologies: controls *vs* bacterial pneumonia *vs* viral pneumonia *vs* COVID-19.

This work skillfully combines a wide range of techniques based on machine learning and sparse representation for image classification, providing a large performance while reducing the computational cost associated with other techniques such as deep learning. The rest of the paper is organized as follows. Section 2 reviews related works for image classification. Section 3 describes the different stages of

the methodology proposed. First, the sparse-coding method is presented. Finally, two different classification algorithms are detailed: Support Vector Machines (SVM) and Random Forest (RF). Afterwards, in Section 4, the applicability of our proposal is evaluated in the diagnosis of different types of pneumonia. Details regarding the dataset are also provided in this section, in addition to an explanation of the preprocessing of the images and a description of the two main experiments conducted. Results are summarized in Section 5 and discussed in Section 6, whereas conclusions and future works are available in Section 7.

2. Related works

The use of intelligent systems for image classification is commonly used in a wide range of scenarios. Ref.[56] proposed a deep-learning approach for construction vehicle detection. Ref.[57] developed an automated classification of indoor air quality by analyzing an electroencephalogram (EEG) signal. According to these results, the EEG signal could be used for automatically control indoor environmental quality changes, reducing the drowsiness and increasing attention. Ref.[58] presented a multi-phase blending method to improve the accuracy of object detectors. In a medical-related context, Ref.[59] proposed a solution for maximizing the visibility of the minority class instance in imbalanced datasets. To do so, they employed soft clustering to detect epilepsy and PD. Other studies have focused on the diagnosis of dyslexia. Specifically, Ref.[60] developed a diagnostic method based on involuntary neurophysiological responses to different auditory stimuli. Briefly, they analyzed the temporal behavior and spectral content of EEG signals to infer a connectivity model that reveals the brain areas that process auditory stimuli in a synchronized way.

With reference to sparse representation, this framework has also been widely used in different contexts such as machine learning,[61,62] computer vision[63,64] and pattern recognition.[65,66] This technique has demonstrated to be an effective way of handling data sets with high dimensionality,[67] and an excellent option for image classification.[68] Ref.[69] proposed a two-layer sparse coding for classifying images obtained from satellites, whereas Ref.[70] introduced a weighted sparse approach for the classification of hyperspectral images. Sparse representations have also been considered for medical image classification. Ref.[71] presented an alternative to K-Means clustering algorithm based on spatial pyramid image representation, demonstrating a high classification performance. Other algorithms focused on preserving the spatial information within the sparse operation, alleviating its instability and improving the precision in the image representation.[72] The high performance of this methodology has also been applied in medical diagnoses. Ref.[73] employed sparse coding for segmenting brain tumors based on their location and intensity. Similarly, Ref.[74] proposed an algorithm based on dictionary learning and sparse coding for the identification of different types of tumors in histological images, whereas Ref.[75] used a very similar framework for segmenting lesions associated with multiple sclerosis. Other previous works have introduced the use of sparse learning methods in the CDR computation from retinal fundus,[76] as well as in cataract grading from slit lamp lens images.[77] Histopathological images are another important context where sparse coding has been employed. Specifically, sparse representations have been used in conjunction with group clustering priors for cervigram segmentation. Similarly, Ref.[78] combined an autoencoder with a dictionary learning framework to extract sparse features in a classification context, whereas Ref.[79] added a spatial pyramid matching to enhance the performance of the system.

3. Methodology

3.1. *Patches extraction*

One crucial aspect for obtaining a good classification performance is related to the number of samples available. The higher this number, the more discriminating features could be learnt by the classifier, increasing the resulting accuracy in most of scenarios.[80,81] Unfortunately, it is not uncommon to have access to datasets in which the number of samples is quite reduced. This is highly established in medical images datasets, since their acquisition is not unexpensive. Another important point to be considered is the existence of inhomogeneities between images of the same class. These images can have similar low-level features, but differing in the location, shape or size of their most relevant characteristics. To overcome these issues, we propose a non-overlapping tile extraction step. Let $I_{H\times W}$ be

the original image, where H and W are its height and width, respectively. The image is then divided into N number of tiles (also known as patches), p_i, with $i = \{1, 2, \ldots, N\}$ and patch dimensions given by $h \times w$. Thus, the number of samples to train the classifier is multiplied by the number of patches divisions (see Figure 1), as follows:

$$N = \left(\frac{H}{h}\right) \times \left(\frac{W}{w}\right) \tag{1}$$

It is worth mentioning that it would be possible to employ an overlapping patches-extraction approach. However, we discard this alternative despite leading to a higher number of samples for the subsequent classification process. The main reason is that overlapping patches contain redundant information for a group of pixels.[82] This is usually controlled by a sliding parameter, which defines the magnitude of the step from a patch to the next one. In an extreme scenario in which this parameter is very small (e.g. s=1), most of pixels contained in one patch are also in the adjacent ones. The use of a non-overlapping approach guarantees the spatial independence of all the subdivisions of the image, preserving the relationship between the informative features and their location.

3.2. *Sparse coding*

After dividing the images into patches, they are entered into a sparse coding process. The idea behind this technique is derived from the way the primary visual cortex in human brain works. The number of neurons in this brain region is much higher than the number of receptor cells in the retina, which suggests that a sparse code is used to efficiently represent natural scenes.[83–85] Sparse coding relies on the assumption that data can be represented in terms of a linear combination of basis elements,[63] which form the so-called dictionary.[86] Consider a number of samples of class i, $A_i = [\mathbf{v}_{i,1}, \mathbf{v}_{i,2}, \ldots, \mathbf{v}_{i,n_i}] \in \mathbb{R}^{m \times n_i}$. A new sample $y \in \mathbb{R}^m$ can be approximated by the linear span of the initial number of samples as follows:

$$\mathbf{y} = \alpha_{i,1}\mathbf{v}_{i,1} + \alpha_{i,2}\mathbf{v}_{i,2} + \cdots + \alpha_{i,n_i}\mathbf{v}_{i,n_i} \tag{2}$$

where $\alpha_{i,j} \in \mathbb{R}, j = 1, 2, \ldots, n_i$ represent the elements of the dictionary, which are usually termed atoms. As its name suggests, the sparsity nature of the sparse coding method relies on the ability of representing a signal in terms of a linear combination of a few atoms. From a mathematical perspective, a solution to the problem can be found by obtaining a matrix such that each column of the input data can be approximated by a linear combination of the columns of the input matrix.

3.3. *Creation of the dictionary*

The main advantage of sparse coding is that a complex signal (e.g. a medical image) can be represented in a very concise manner. A crucial aspect in this process is how the dictionary is built. The simplest alternative is to use a prespecified transform matrix, which usually leads to fast solutions for the evaluation of the sparse representation. One option is to generate a dictionary entry for each individual patch of all images. This would lead to a dictionary size given by the number of samples available (number of patches $\times$ number of images). It can be problematic that the size of the resulting dictionary is too high for two main reasons. First, a dictionary with a high number of atoms would lead to a slow process when transforming and reconstructing the images because of the matrices multiplication this process relies on. Second, employing features in the original space can be suboptimal, making that atoms do not reflect the main features of the different classes. In this work, we propose a method based on Principal Component Analysis (PCA) for the creation of the dictionary in order to maximize the differences in the representation of the different images.[16] Briefly, patches of different images are individually stored by columns in a matrix, and PCA[87–89] is then applied to this matrix. The aim of PCA is to find the projecting directions of maximum variance of a certain subspace given N samples $\mathbf{x}_k$, with $\mathbf{x}_k = [\mathbf{x}_{k1}, \ldots, \mathbf{x}_{kn}] \in \mathbb{R}^n$.[90] This means that the vector, $\mathbf{x}$, is projected from the original space, $\mathbb{R}^n$, to a new one with a higher dimension, $\mathbb{R}^f$. The eigenvalue problem in the new feature space is given by:

$$C^{\Phi}\mathbf{w}^{\Phi} = \lambda\mathbf{w}^{\Phi} \tag{3}$$

where the covariance matrix is represented by C^{Φ}. The transformed space, $\Phi(\mathbf{x}_1, \ldots, \Phi(\mathbf{x}_N))$, contains the solutions $\mathbf{w}^{\Phi}$ with $\lambda \neq 0$. Regarding coefficients

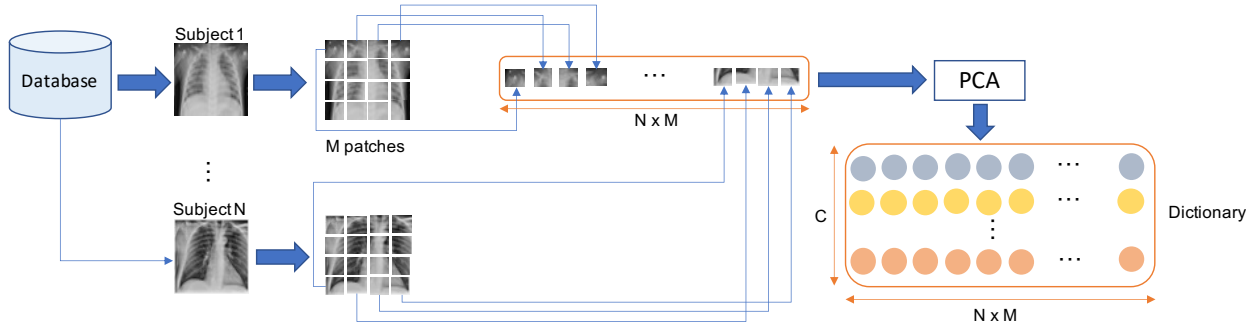

Figure 1. Process for the creation of the dictionary. First, images are divided into patches of equal size. These patches are organized by columns into a matrix. A PCA is then used to maximize the variance of the projected components. The resulting ones form the final dictionary.

β, they are given by:

$$\mathbf{w}^{\Phi} = \sum_{i=1}^{N} \beta_i \Phi(\mathbf{x}_i) \tag{4}$$

Each element of matrix K, with a NxN size, can be defined as:

$$K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) \tag{5}$$

Thus, the mathematical problem is redefined as:

$$N\lambda K\boldsymbol{\beta} = K^2\boldsymbol{\beta} \rightarrow N\lambda\boldsymbol{\beta} = K\boldsymbol{\beta} \tag{6}$$

where $\boldsymbol{\beta}$ is a vector with elements $\beta_1 \dots \beta_N$.[91]

Finally, vectors in the new feature space are projected into a space with a lower dimension described by the eigenvectors $\mathbf{w}^{\Phi}$. Given a sample $\mathbf{x}$, $\Phi(\mathbf{x})$ would be its projection in the transformed space. According to Equation 4 and Equation 5, the projection of $\Phi(\mathbf{x})$ onto the eigenvectors $\mathbf{w}^{\Phi}$ corresponds to the nonlinear principal components of Φ, as follows:

$$\mathbf{w}^{\phi} \cdot \Phi(\mathbf{x}) = \sum_{i=1}^{N} \beta_i (\Phi(\mathbf{x}_i)\Phi(\mathbf{x})) = \sum_{i=1}^{N} \beta_i K(\mathbf{x}_i, \mathbf{x}) \tag{7}$$

Figure 1 depicts a schematic representation of the dictionary generation. The eigenvectors (also known as principal components) of the covariance matrix correspond to the directions of maximum variance of the data manifold. Given that eigenvectors are obtained from images, they are useful for determining the variance of patterns associated with the different classes contained in a collection of images. Since the first eigenvectors compress the most part of the explained variance, they are usually chosen in increasing-variance order. Thus, the resulting first C components are the atoms that form the dictionary, guaranteeing that these atoms represent crucial information to be used in the subsequent classification. We include in Algorithm 1 a pseudocode to clarify the construction of the dictionary.

Algorithm 1 Pseudocode for the dictionary construction.

```
procedure GenerateDictionary(steps)
    // Input: CXR Images
    for i = 1 → NSubjects do
        img = Data[i,:]
        // Image is divided into P patches
        img = patchify(img)
        // Create a matrix of F files (number of pixels in each patch) and NSubjects x P columns
        patches = zeros(F, NSubjects * P)
        a = 1
        for j = 1 → P do
            patches[:,a] = img[j]
            a = a+1
        end for
    end for
    eigenvectors = PCA(patches)
    dictionary = eigenvectors
    // Compute sparse representation of patches
    sparse_patches = dictionary(patches)
    // Create a matrix for reconstructed patches
     rec_patches = zeros(F, NPatches)
    // And another one for the reconstruction errors
    rec_errors = zeros(NComponents, NPatches)
    a=0
    for k = 1 → NPatches do
        // Deactivate atoms from Component c
        dictionary[:,c] = 0
        rec_patch = dictionary(sparse_patches)
        rec_errors[c,k] = patch-rec_patch
    end for
    //Divide rec_errors in training and test
    model.fit(rec_errors_tr)
    pred = model.predict(rec_errors_te)
end procedure
```

3.4. *Sparse representation*

Once the dictionary is built, the sparse representation of the different patches is computed. Specifically, the decomposition coefficient vector $\hat{\alpha}$ is obtained by solving the L1-norm minimization[92,93] problem as follows:

$$\hat{\alpha} = \arg\min_{\alpha} \|\alpha\|_1 \text{ subject to } \|\mathbf{X}\alpha - \mathbf{y}\|_2 \leq \epsilon \quad (8)$$

where $\mathbf{X}$ is the dictionary and $\mathbf{y}$ is the image patch to be transformed. All patches are reconstructed by employing a modified version of the dictionary within an iterative process in order to evaluate the information of each individual class stored by each component. First, the atoms associated with the first component are set to zero, using the elements of all but this component. In the second step, the unused atoms are the ones related to the second component, and so on. The number of reconstructions for each individual patch is given by the number of components employed during the construction of the dictionary. Then, a sparse reconstruction error is estimated for each component. These errors are obtained by comparing the reconstruction of each patch with the original patch. Specifically, for each individual component is deactivated, the difference between the original patch and the reconstructed version is computed, as follows:

$$e = \|\mathbf{X}\hat{\alpha}\| - \mathbf{y} \quad (9)$$

Previous studies have employed the reconstruction errors for deciding the class each test sample belongs, in a process known as sparse coding classification (SRC).[63,86,92,94] In this approach, the reconstruction errors are computed by employing the sparse coefficients $\hat{\alpha}^l$ associated with each class l:

$$e_l = \left\|\mathbf{X}^l\hat{\alpha}^l\right\| - \mathbf{y} \text{ for } l = 1, \ldots, C \quad (10)$$

where C is the number of classes. The class label for the test sample $\mathbf{y}$ is assigned as the class with the minimum reconstruction error:

$$Label(y) = \arg\min_{l} e_l(y) \quad (11)$$

The simplicity of SRC is detrimental in scenarios where informative patterns are very similar in the different classes. Moreover, our framework does not lead to an only reconstruction error, but a number given by the components employed when constructing the dictionary and the patches contained in each image. Thus, the use of SRC within our scheme would lead to a different class assignment for each patch and component, which could be then combined into a global classification decision (e.g., with an ensemble approach[95,96]). However, we designed our method as a feature extractor prior to the classification task, employing a more complex algorithm to separate the different classes.

3.5. *Classification*

3.5.1. *Support Vector Machines*

The reconstruction errors were then entered as input of the classification stage, which was based on an SVM classifier with a linear kernel. This algorithm selects a hyperplane that separates the samples from the two classes. A linear SVM classifier, f, can be mathematically described by a pair of $(\mathbf{x}, \mathbf{x})$, as follows:

$$f(\mathbf{x}_i) = \langle \mathbf{w}, \mathbf{x}_i \rangle + b \tag{12}$$

where $\mathbf{w}$ and $\mathbf{x}_i$ are the weight and the feature vector, respectively, and b is the error term. A point x would be classified as positive or negative depending on if $f(x) > 0$ or if $f(x) < 0$. The resulting hyperplane with the maximum margin is computed by solving the optimisation problem described in Ref.[97] :

$$\begin{aligned} &\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i \xi_i \qquad \text{subject to} \\ &y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i \qquad \forall_i \xi_i \geq 0 \ \ \forall_i \end{aligned} \tag{13}$$

where C is the penalty for misclassification. The optimisation problem can be solved as follows:

$$\mathbf{w} = \sum_{i=1}^{n} y_i \alpha_i \mathbf{x}_i \tag{14}$$

once Lagrangian multipliers are applied. The decision function can be rewritten in its dual form by substituting the value of $\mathbf{w}$ in Equation 12:

$$f(\mathbf{x}_i) = \sum_{i=1}^{n} \alpha_i K(\mathbf{x}, \mathbf{x}_i) + b \tag{15}$$

where α_i and b are the coefficients to be learnt from the examples and $K(\mathbf{x}, \mathbf{x}_i)$ is the kernel function employed to characterize the similarity between samples $\mathbf{x}$ and $\mathbf{x}_i$. Since classes were unbalanced, we included the weights of the different classes into the cost function in order to both classes contribute equally.

3.5.2. *Random Forest*

For a multiclass classification, the reconstruction errors of the different patches of the images were entered into a Random Forest classifier. RF is an ensemble method that combines a number of decision trees in order to improve the performance of individual classifiers. The trees are built from k random vectors, $\mathbf{\Theta}_k$, which are independent of the past random vectors $\mathbf{\Theta}_1, \mathbf{\Theta}_2, \mathbf{\Theta}_3, \ldots, \mathbf{\Theta}_{k-1}$ but with the same distribution. The process developed in Ref.[98] employs bagging for generating each random vector $\mathbf{\Theta}$ as the N observations randomly drawn from the training set. Once a large number of trees is generated $\{h(\mathbf{x}, \Theta_k), k = 1, \ldots N\}$, each one of them casts a vote for the most popular class at input $\mathbf{X}$. The majority vote of the trees determines the final decision of the ensemble.

One important aspect of RF classifiers is related to their convergence and generalization error.[98,99] Given a set of classifiers $h_1(\mathbf{x}), h_2(\mathbf{x}), \ldots, h_k(\mathbf{x})$, and a training set randomly drawn from the distribution of a random vector (X,Y), the margin function is defined as:

$$mg(\mathbf{X}, Y) = av_k I(h_k(\mathbf{X}) = Y) - max_{j \neq Y} av_k I(h_k(\mathbf{X}) = j) \tag{16}$$

where $\mathbf{X}$ is the input metric, av_k refers to the average number of votes at $\mathbf{X}, Y$ for the corresponding class and $I(\cdot)$ is the indicator function. The margin is a measure about the extent to which the average number of votes at $\mathbf{X}, Y$ for the right class exceeds the average vote for any other class. Thus, the larger the margin, the more confidence in the classification. The generalization error es given by:

$$PE^* = P_{\mathbf{X},Y}(mg(\mathbf{X}, Y) < 0) \tag{17}$$

where $P_{\mathbf{X},Y}$ indicates that the probability is over the $\mathbf{X}, Y$ space. According to Theorem 1.2 in Ref.,[98] as the number of trees increases, for almost surely all sequences $\Theta_1, \ldots PE^*$ converge to:

$$P_{\mathbf{X},Y}(P_\Theta(h(\mathbf{X}, \Theta) = Y) - max_{j \neq Y} P_\Theta(h(\mathbf{X}, \Theta) = j) < 0) \tag{18}$$

Figure 2 shows a schematic representation of the entire classification framework, from the dictionary creation within the sparse coding to the classifier's decision. This process can be summarized as follows:

- Division of the images into patches. Informative patterns can be located in small regions, so that partitioning the images increases the number of exemplars of informative and non-informative regions. The size of the resulting

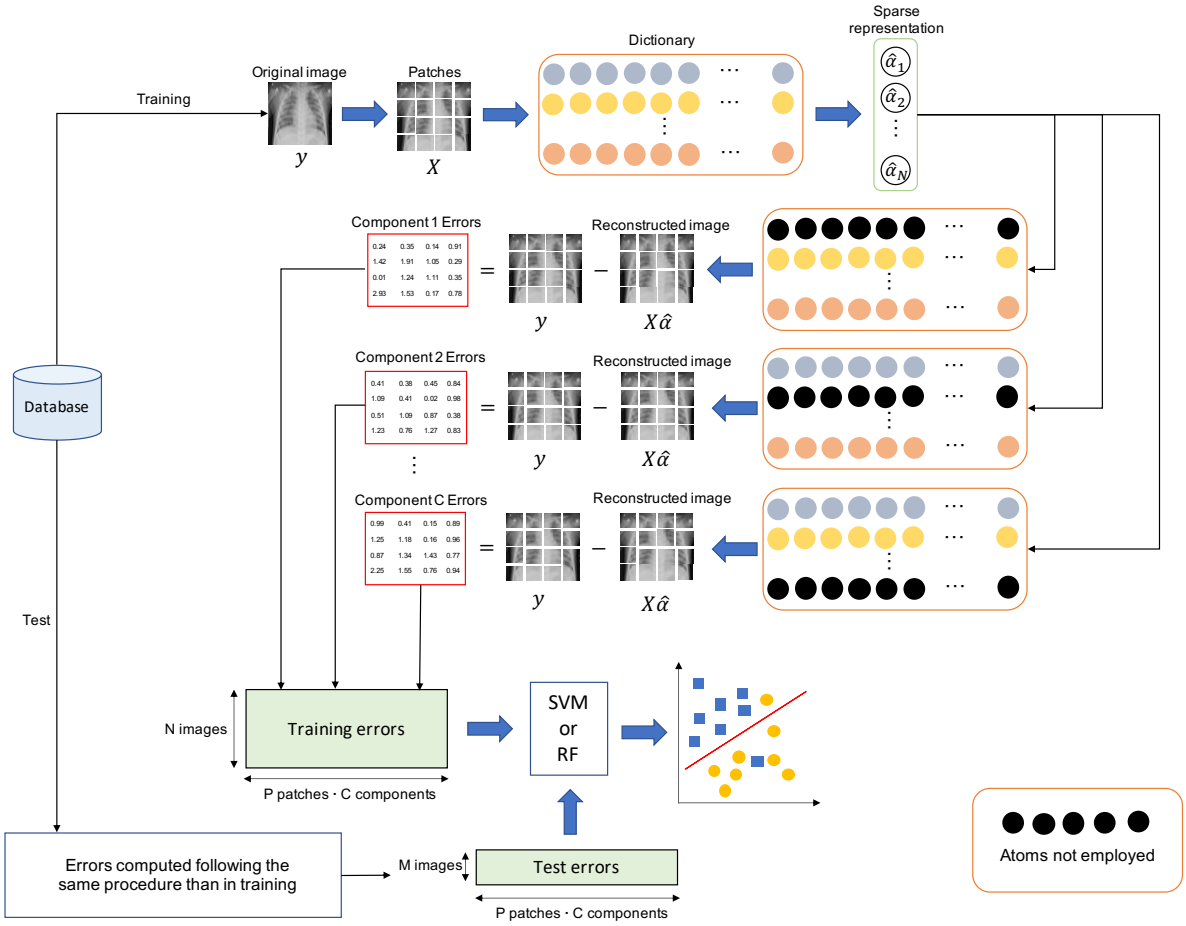

Figure 2. Scheme of the classification pipeline. After building the dictionary, the sparse representation of each image patch is computed. The reconstruction is performed by employing an iterative process in which the elements of the dictionary are partially employed. First, the atoms associated with the first component are set to zero, using the elements of all but this component. In the second step, the unused atoms are the ones related to the second component, and so on. This process is performed both in the training and the test set. The reconstruction errors are computed as the difference between the original and the transformed images. The resulting errors obtained in each individual patch are then used as the input of the classifier, leading to the label prediction.

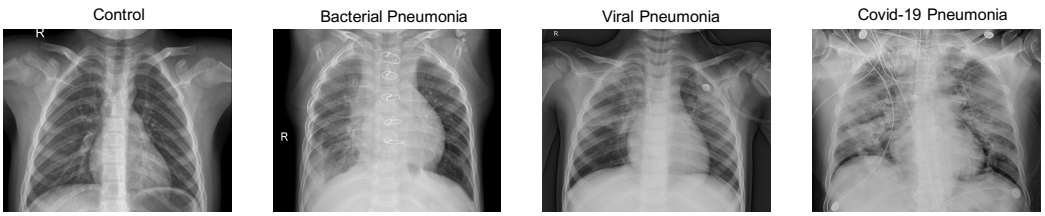

Figure 3. CXR images of the different pathologies evaluated in this work.

patches must be adapted to the specific characteristics of the images to be processed. For example, the size of the patches in pneumonia images must be large enough to contain the patterns of the pulmonary damage associated with this pathology.

- Creation of a matrix where each column corresponds to an individual patch. The size of the resulting matrix will be $M \times N$, where M is the number of voxels contained in each patch and N is the product of the number of patches contained in each image and the total number of images.
- Application of PCA to this matrix to obtain an optimum dictionary by maximizing the variance of the projected components. The eigenvectors will be the elements of the dictionary, so that their size will be derived from the number of principal components preserved.
- Computation of the decomposition coefficient vector $\hat{\alpha}$ by solving the L1-norm minimization problem by sparse coding: $\hat{\alpha} = \arg\min_{\alpha} \|\alpha\|_1$
- Estimation of the reconstruction error for each patch $\mathbf{y}$. To do so, the sparse coefficients associated with each component are iteratively set to zero to evaluate their influence in the reconstruction performance. Each error is computed as follows: $e = \|\mathbf{X}\hat{\alpha}\| - \mathbf{y}$
- The resulting reconstruction errors are the input features of a SVM/RF classifier in case of a binary/multiclass classification, respectively.

4. Application to pneumonia detection

In the following subsections, the proposed methodology is evaluated in detail using chest X-ray (CXR) images to show the applicability of the proposal in the detection of pneumonia. Specifically, we evaluated the performance of our method in a real context where distinguishing between controls and different types of pneumonia: bacterial, viral and COVID-19.

4.1. *Database description*

Images from controls and bacterial/viral pneumonia were obtained from Ref.[100] . These images were acquired from retrospective cohorts of one to five years patients from Guangzhou Women and Children's Medical Center, Guangzhou[101] as part of routines clinical care. Institutional Review Board (IRB)/Ethics Committee approvals were obtained. The work was conducted in a manner compliant with the United States Health Insurance Portability and Accountability Act (HIPAA) and was adherent to the tenets of the Declaration of Helsinki. The dataset comprises 5856 CXR images, 4273 from pneumonia patients and 1583 controls. The COVID-19 dataset contains 576 CXR images from adults.[102] Figure 3 shows the CXR images associated with the different pathologies evaluated in this study: controls (CTL), bacterial (BAC) and viral (VIR) pneumonia, and COVID-19 (CVD-19).

4.2. *Image preprocessing*

CXR are usually images with a low resolution and high levels of noise. This is mainly due to the low X-ray radiation applied during the acquisition of the images as well as the movement of patients along this process. Preprocessing can mitigate these effects and improve the results of the subsequent classification. We first downsampled the images to a final size of 224x224 in order to mitigate the computational burden while preserving their quality. Then, an intensity normalization based on standardization was applied. This procedure transforms each image in order to the resulting distribution has a mean (μ) of 0 and a standard deviation (σ) of 1:

$$I' = \frac{I_o - \mu}{\sigma} \tag{19}$$

where I_o and I' are the original and the preprocessed image, respectively.

4.3. *Sparse coding representation from patches*

As explained in previous sections, we propose the use of sparse coding to obtain a sparse representation of an input image from a linear combination of elements. In our case, the elements correspond to the atoms of a dictionary, obtained after applying PCA to a matrix where patches of different images were contained. This allows that the atoms maximize the differences in the representation of the different classes. In this process, there are two main parameters to have into account in the experiments.

- The number of patches each image is divided into.
- The threshold on the variance explained, which limits the number of eigenvectors used in the PCA projections.

In the experiments, we varied the size of the patches and the number of resulting components from PCA to evaluate their influence in performance.

4.4. *Performance evaluation*

In order to estimate the generalization ability of the classifier, we used a 5-fold stratified cross-validation scheme for all experiments.[103] With reference to performance, we computed the following parameters derived from the confusion matrix:

$$Bal\ Acc = \frac{1}{2}\left(\frac{TP}{P} + \frac{TN}{N}\right) \qquad Prec = \frac{T_P}{T_P + F_P}$$

$$Sens = \frac{T_P}{T_P + F_N} \qquad Spec = \frac{T_N}{T_N + F_P}$$

$$F1 - score = \frac{2 \times Prec \times Sens}{Prec + Sens} \quad AUC = \frac{1}{2}\left(\frac{TP}{P} + \frac{TN}{N}\right)$$

where T_P corresponds to the number of patients correctly classified as pneumonia (true positives), T_N refers to the number of controls properly identified (true negatives), F_P quantifies the number of controls labelled as pneumonia (false positives), whereas F_N refers to false negatives, i.e. the number of pneumonia patients who are mistakenly diagnosed as controls. We employed as an additional measure the area under the ROC curve (AUC).[104,105] In the multiclass scenario, the information derived from parameters such as *Sens* or *Spec* can not be easily interpreted. In this context, we employed a method based on a multi-class One-vs-One scheme to compare every unique pairwise combination of classes.[106] The multiclass-AUC was computed by averaging the results obtained for each individual comparison. Moreover, a multiclass version of the balanced accuracy was computed, as follows:

$$Multiclass\ Bal\ Acc = \frac{1}{M}\sum_{m=1}^{M}\frac{r_m}{n_m} \qquad (20)$$

where M is the number of classes, n_m is the number of samples belonging to class m and r_m is the number of samples belonging to class m that are accurately predicted.

4.5. *Experimental setup*

We define three different experiments associated with the identification of pneumonia patterns:

- **Experiment 1: Binary Classification** to differentiate between the different pathologies in three contexts: **CTL *vs* PNEU**, which includes all images labelled as CTL and PNEU regardless of the type of pneumonia; **BAC *vs* VIR**, which divides the images from patients diagnosed from pneumonia according to the cause of the disease (bacterial or viral); **VIR *vs* CVD19** for viral pneumonia. In the last context, the aim was to identify whether viral pneumonia was caused by COVID-19 or not. In this first experiment, the resulting features from the sparse coding phase were then entered into a linear SVM classifier. The cost parameter was optimized within a grid-search process in the training phase, ranging from 10^{-5} to 10^{5}.
- **Experiment 2: Multiclass Classification** by using an RF classifier in order to distinguish between the four different pathologies contained in the database. This algorithm combines the decisions of individual trees to obtain the final diagnosis of the patient. The process for building the dictionary in addition to the classification framework are identical to Experiment 1 except the aforementioned change in the classifier.
- **Experiment 3: Evaluation of the effect of different parameters in the classification performance.** First, we repeated the previous experiments with a limited number of samples in order to assess how performance varies when a lower number of samples is available. Specifically, the size of the dataset was iteratively reduced from 100% to 25% with a step size of 25%. Besides, two different kernels (polynomial of second and third degree and Radial-Basis Function (RBF)) were used in order to compare the original results in the four classification frameworks where a linear kernel was employed.

The code was written in Python 3.6, and a number of additional libraries was used: Scikit-Learn 1.0, Numpy 1.19.5, Patchify 0.2.3. The experiments were carried out on a cluster with the following hardware specifications: two Intel® Xeon® E5-2630 node

Table 1. Summary of previous works focused on the automatic identification of pneumonia in addition to the best results obtained by our method.

Research work	Dataset	Method	Classification context	Results (%)
[107]	1000 CT scans	GAN model	Normal *vs* COVID	Acc = 99.95
[108]	4356 CT scans	COVNet	Normal *vs* COVID	AUC = 96.00
[109]	6374 CXR images	Bayesian Deep Learning	Normal *vs* Bacterial *vs* Viral *vs* COVID pneumonia	Acc = 98.06
[110]	513 CT scans	Probabilistic Machine Learning	Normal *vs* COVID	Acc = 97.86
[111]	137 CT scans	3D-Resnet-10	Severe *vs* Critical COVID	AUC = 90.90
[112]	2905 CXR images	Fibonacci patterns	Normal *vs* COVID	Acc = 99.78
[113]	13962 CXR images	DeepDRR	Normal *vs* COVID	Acc = 94.00
[114]	3141 CXR images	Resnet-50	Normal *vs* COVID	Acc = 96.10
[115]	3487 CXR images	DenseNet-201	Normal *vs* Bacterial *vs* Viral pneumonia	Acc = 97.94
[116]	1428 CXR images	VGG19	Normal *vs* COVID *vs* Bacterial pneumonia	Acc = 98.75
[117]	3993 CXR images	Resnet-50	Normal *vs* COVID *vs* Other pneumonia	Acc = 99.87
[118]	852 CXR images	COVIDNet	Normal *vs* COVID	Acc = 97.72
[119]	3487 CXR images	CheXNet	Normal *vs* Bacterial *vs* Viral pneumonia	Acc = 97.80
[120]	1142 CXR images	DarkNet	Normal *vs* COVID *vs* Viral pneumonia	Acc = 87.02
[121]	400 CT scans	VGG16	Normal *vs* COVID	Acc = 99.00
[122]	234 CT scans	DenseNet-121	Normal *vs* COVID	Acc = 99.00
[123]	1110 CT scans	COV-CAF	Normal *vs* COVID	Acc = 97.76
[124]	1164 CT scans	CCSHNet	Normal *vs* COVID *vs* Pneumonia *vs* Tuberculosis	Acc = 96.46
[125]	4886 CXR images	Ensemble Deep Learning	Normal *vs* COVID	Acc = 99.80
[126]	63849 CT scans	ResNet-50V2	Normal *vs* COVID	Acc = 99.49
Our method	**6432 CXR images**	**Sparse coding**	**Normal *vs* Pneumonia**	**AUC = 97.39**
Our method	**4849 CXR images**	**Sparse coding**	**Bacterial *vs* Viral pneumonia**	**AUC = 84.51**
Our method	**2069 CXR images**	**Sparse coding**	**Viral *vs* COVID19 pneumonia**	**AUC = 99.14**
Our method	**6432 CXR images**	**Sparse coding**	**Normal *vs* Bacterial *vs* Viral *vs* COVID pneumonia**	**AUC = 87.04**

2.40GHz processors, with 10 cores per processor. Besides, the total RAM memory capacity of the system is 128 GB.

Table 1 provides an overview of recent works focused on the automatic detection of pneumonia, including the methodology employed and the results obtained. Despite the number of works for the detection of pneumonia is high, comparing between different approaches is not a straightforward task because two main reasons. First, many studies employ datasets that are not publicly available. Since results highly depend on the difficulty of the classification task, they are affected by the images used. Second, there is a high variability between the datasets that most works use: from the modality of the images (CT or X-ray), to the pathologies to be detected (viral/bacterial pneumonia, COVID, other lung abnormalities, etc) or the number of images from each class.

5. Results

We first explore how performance varies according to two parameters: the number of patches each image is divided into and the number of components retrieved from PCA to build the dictionary. Results are summarized in Table 2 for the four different classification contexts. We can see that the maximum accuracy obtained in the CTRL *vs* PNEU scenario is 97.74%, with a patch size of 14x14 and 9 components used to compute the dictionary. It is important to note that there is not a clear relationship between these two variables and the resulting accuracy. However, a drop in accuracy appears when too large patches are used (56x56). This can be related to the fact that pneumonia patterns are usually located in small regions of the CXR images. When applying sparse coding, information extracted can be related to pulmonary affections derived from pneumonia. However, when the size of patches increases, this information can be due to other sources such as pulmonary structures that are completely normal, increasing the difficulty of the classification task.

It is important to mention that the performance in the second context (BAC *vs* VIR pneumonia) is slightly lower than in the first scenario, manifesting the higher difficulty of this classification. Specifically, the maximum accuracy was 84.88%, with a patch size of 14x14 and 9 components. We also observe that the discrimination ability of the proposed system is larger in the VIR *vs* CVD19 scenario, with a maximum accuracy of 99.36%. This can evidence that the pathology caused by COVID-19 is more severe and different than the one caused by other virus or bacteria. Finally, the best result in the multiclass context led to an accuracy of 86.73%. Results in terms of different metrics associated with the situation of max-

Table 2. Balanced accuracies (and their deviations) of the classification approach proposed in this work in the different contexts evaluated. Patch size is given in pixels.

Controls *vs* Pneumonia								
	Number of PCA components							
Patch size	N=2	N=3	N=4	N=5	N=6	N=7	N=8	N=9
14x14	96.13±0.35	95.79±0.59	96.34±0.44	96.91±0.49	97.43±0.59	97.65±0.41	97.05±0.31	**97.74±0.33**
16x16	95.96±0.54	95.85±0.43	95.99±0.53	96.59±0.31	96.73±0.25	96.86±0.24	96.88±0.33	96.77±0.43
28x28	93.95±1.17	94.22±0.97	93.80±0.83	94.02±1.01	94.06±0.76	93.22±0.88	93.58±0.45	93.63±0.54
32x32	89.63±1.52	92.97±0.72	92.64±1.02	92.14±0.86	91.92±0.65	92.02±1.37	92.26±0.76	92.96±0.73
56x56	76.29±1.72	81.95±0.30	82.93±0.75	84.69±0.61	86.75±1.19	87.45±1.24	86.48±0.81	87.90±0.76
Bacterial *vs* Viral pneumonia								
	Number of PCA components							
Patch size	N=2	N=3	N=4	N=5	N=6	N=7	N=8	N=9
14x14	83.29±0.98	82.76±0.95	84.16±0.79	83.85±1.16	84.98±0.98	84.24±1.43	83.87±0.47	**84.88±1.13**
16x16	82.83±1.29	81.74±0.92	82.48±0.91	81.29±0.64	84.55±1.13	82.54±1.13	83.39±1.48	83.78±0.65
28x28	79.57±1.25	79.43±1.42	79.76±1.09	79.84±1.21	80.67±1.40	79.84±1.38	79.28±2.09	80.29±1.21
32x32	79.33±1.10	79.53±1.15	79.43±1.63	80.17±1.56	79.38±0.99	80.00±1.60	79.55±0.94	79.20±1.40
56x56	61.60±1.26	67.96±1.59	69.63±1.84	74.08±1.45	74.06±1.17	73.89±1.27	75.64±1.27	75.94±1.00
Viral *vs* COVID19 pneumonia								
	Number of PCA components							
Patch size	N=2	N=3	N=4	N=5	N=6	N=7	N=8	N=9
14x14	99.08±0.01	99.17±0.02	98.24±0.02	98.75±0.01	98.56±0.01	98.78±0.01	98.96±0.01	99.02 ±0.01
16x16	98.87±0.01	98.86±98.86	98.92±0.01	98.92±0.01	98.90±0.01	**99.36±0.01**	99.14±0.01	98.98±0.01
28x28	97.81±0.01	98.23±0.01	98.75±0.02	98.86±0.01	98.96±0.01	98.95±0.01	98.91 ±0.01	98.95±0.01
32x32	97.47±0.02	98.13±0.05	98.76±0.01	99.76±0.01	98.85±0.01	98.55±0.01	98.95±0.01	98.95±0.01
56x56	96.52±0.03	98.36±0.05	98.79±0.05	98.74±0.04	98.84±0.02	97.08±0.03	98.23±0.04	98.51±0.03
Multiclass								
	Number of PCA components							
Patch size	N=2	N=3	N=4	N=5	N=6	N=7	N=8	N=9
14x14	85.05±1.40	85.41±1.19	85.98±1.04	85.54±1.07	86.59±0.92	**86.73±1.08**	84.61±1.08	84.16±1.30
16x16	85.37±1.15	84.22±1.11	84.50±1.33	84.05±1.23	84.31±1.22	84.64±0.99	85.14±1.09	83.44±1.10
28x28	82.83±1.06	81.81±1.29	82.21±1.20	81.27±1.26	81.59±1.17	81.61±1.21	81.28±0.87	81.46±1.13
32x32	82.33±1.45	81.49±1.42	81.02±1.19	80.79±1.38	81.62±1.27	81.17±1.04	80.82±1.02	80.71±0.99
56x56	67.21±1.21	72.43±0.80	74.72±1.18	75.19±0.75	74.56±0.67	74.75±0.84	73.97±0.84	75.49±0.97

Table 3. Performance metrics obtained in the maximum balanced accuracy scenario for all the classification contexts evaluated.

Controls *vs* Pneumonia					
Bal Acc (%)	Sens (%)	Spec (%)	AUC (%)	Prec (%)	F1-score (%)
97.74±0.33	98.08±0.33	96.70±1.29	97.39±0.60	98.92±0.42	98.50±0.22
Bacterial *vs* Viral Pneumonia					
Bal Acc (%)	Sens (%)	Spec (%)	AUC (%)	Prec (%)	F1-score (%)
84.88±1.13	81.97±1.13	87.05±1.66	84.51±1.07	82.57±1.91	82.26±1.21
Viral *vs* COVID19 Pneumonia					
Bal Acc (%)	Sens (%)	Spec (%)	AUC (%)	Prec (%)	F1-score (%)
99.36±0.02	99.05±0.01	99.56±0.01	99.14±0.02	98.72 ±0.01	98.88±0.02
Multiclass					
Multiclass Bal Acc (%)			AUC (%)		
86.73±1.08			87.04±1.21		

imum accuracy are shown in Table 3. Figure 4 summarizes the influence of the size of each individual tile in the classification performance. The maximum accuracies are obtained with squared patches of 14x14 or 16x16 pixels in the different classification contexts. Our results show that the accuracy starts decreasing when too large patches are employed. This evidences that covering too wide regions can be detrimental for the identification of pneumonia, especially in cases where this affection is not severe. Figure 5 depicts the ROC curves for the different classifiers. The best results are obtained in the VIR vs CVD-19

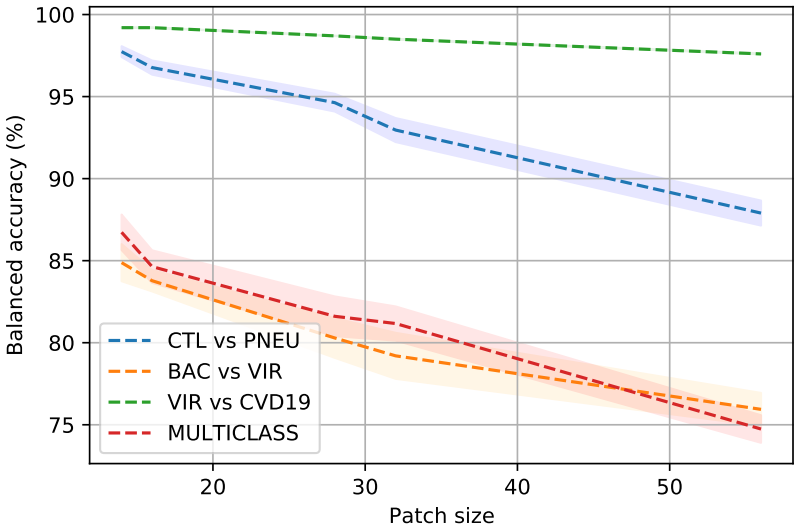

Figure 4. Influence of the patch size in the classification performance for the different contexts evaluated. Representations correspond to the scenario that leads to the maximum performance.

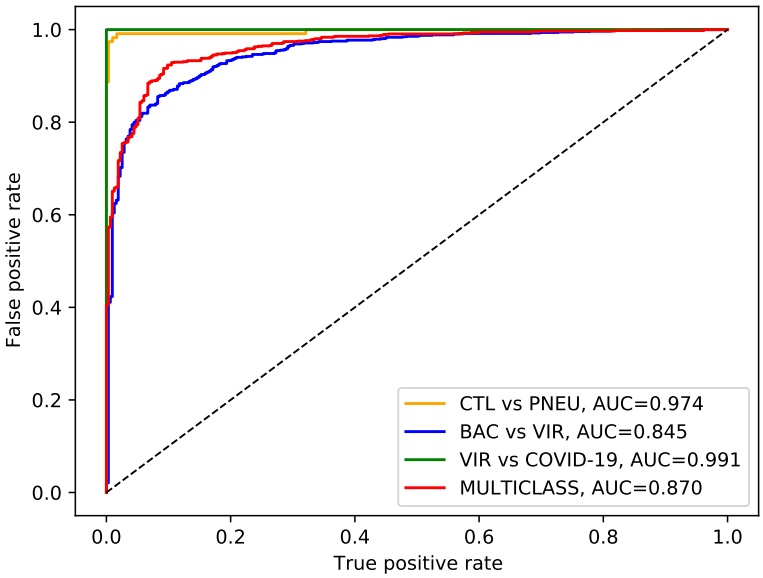

Figure 5. ROC curves obtained in the different classification scenarios. A multiclass version of the ROC curve was computed for the multiclass context.

context, since differences between these two groups of patients are clear. However, our system can also distinguish between patients with the same pathology (pneumonia) but different etiology (bacteria, virus or COVID-19). Figure 6 shows a distribution of the reconstruction errors in the four classification contexts. We can see that errors vary depending on the components used for the image reconstruction. It is also important to note the differential influence of each component when reconstructing patches associated

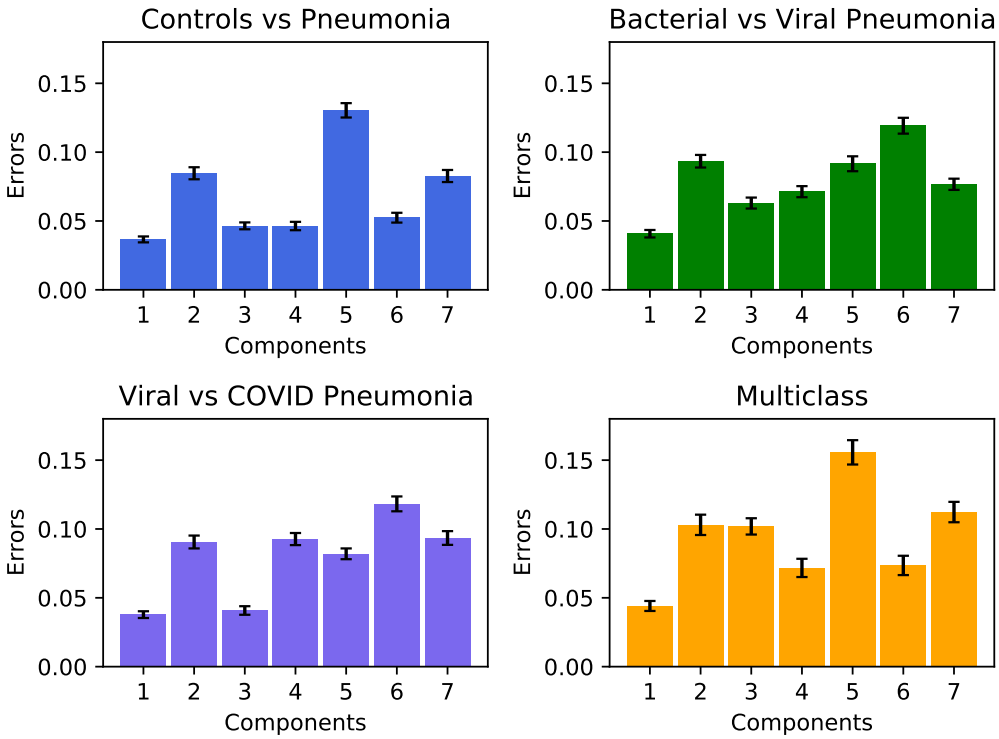

Figure 6. Distribution of the reconstruction errors in the four classification contexts.

with the four classes, which manifests the subsequent discrimination ability of the classifier.

Table 4 summarizes the influence of the size of the dataset in the classification performance. We can see that a decrease in the number of samples does not lead to a drop in performance. Moreover, variations in the results are minimum, leading to slight better performance in some cases when reducing the dataset. This evidences the robustness of our method and its suitability in situations where large datasets are not available. Regarding the effect of the kernel employed, our results show that the linear kernel is the optimum one for most classification scenarios, although performance is very similar for the different kernels (see Table 5). Further discussion about the results and their implications are provided in Section 6.

6. Discussion

In this work, a classification framework based on sparse coding is proposed. This approach is based on the construction of a dictionary that relies on the assumption that an image can be expressed as a linear combination of different atoms. We employed a scheme in which each image was divided into patches and the dictionary was built from the components of maximum variance from the patches of all images. The errors obtained from the iterative reconstruction were then used as input features of a classifier. The performance of this alternative was evaluated in three scenarios. In the first and in the third one, the two classes generated relatively big differences in the observed pattern (pneumonia *vs* control, viral pneumonia *vs* COVID-19), whereas in the second (bacterial *vs* viral pneumonia) these differences were extremely small. Besides, the performance of a multiclass classifier was also evaluated in order to check if this method could simultaneously differentiate between the different pathologies.

Previous studies have employed sparse coding for the processing and analysis of different signals.[16,127] However, most of them have used it within the classification stage instead of as a feature extractor. Specifically, images are reconstructed from atoms of the dictionary corresponding to the different classes. The final label is assigned according to

Table 4. Effect of the sample size in the classification performance. All analyses were carried out with the optimum patch size, i.e. the one that led to a large performance in each clasification when the whole dataset was used.

Controls *vs* Pneumonia								
	Number of PCA components							
Dataset size (%)	N=2	N=3	N=4	N=5	N=6	N=7	N=8	N=9
100	96.13±0.35	95.79±0.59	96.34±0.44	96.91±0.49	97.43±0.59	97.65±0.41	97.05±0.31	97.74±0.33
75	96.57 ±0.86	96.76 ±0.44	97.67 ±0.20	97.46 ±0.55	97.94 ±0.66	98.25 ±0.50	98.11 ±0.56	98.50 ±0.30
50	97.26 ±0.40	97.19 ±0.52	98.17 ±0.39	98.19 ±0.58	98.82 ±0.16	98.75 ±0.10	98.60 ±0.33	98.85 ±0.29
25	97.00 ±0.32	97.51 ±0.65	98.19 ±0.50	98.38 ±0.73	98.69 ±0.36	98.81±0.24	98.44 ±0.34	98.69 ±0.46
Bacterial *vs* Viral pneumonia								
	Number of PCA components							
Dataset size (%)	N=2	N=3	N=4	N=5	N=6	N=7	N=8	N=9
100	83.29±0.98	82.76±0.95	84.16±0.79	83.85±1.16	84.98±0.98	84.24±1.43	83.87±0.47	84.88 ±1.13
75	84.54 ±1.43	85.65 ±0.86	85.23 ±1.14	85.21 ±1.37	85.59 ±0.82	86.31±0.96	87.52 ±0.74	87.69 ±0.81
50	84.92 ±2.26	86.62 ±2.60	86.82 ±2.44	87.64 ±1.98	88.68 ±1.51	89.38 ±1.03	88.39 ±0.99	87.23 ±1.80
25	84.55 ±1.42	85.65 ±0.86	85.21 ±1.11	85.18 ±1.44	85.81 ±0.82	86.25 ±0.99	86.44 ±1.01	86.97 ±0.84
Viral *vs* COVID19 pneumonia								
	Number of PCA components							
Dataset size (%)	N=2	N=3	N=4	N=5	N=6	N=7	N=8	N=9
100	99.08±0.01	99.17±0.02	98.24±0.02	98.75±0.01	98.56±0.01	98.78±0.01	98.96±0.01	99.02 ±0.01
75	98.54 ±0.04	98.76 ±0.07	99.01 ±0.04	99.06 ±0.09	98.89 ±0.06	99.21 ±0.01	99.17 ±0.02	99.09 ±0.01
50	99.05 ±0.06	98.42 ±0.08	98.76 ±0.11	99.14 ±0.08	99.11 ±0.08	99.27 ±0.01	99.25 ±0.02	99.19 ±0.01
25	98.45 ±0.12	98.58 ±0.07	98.21 ±0.14	98.47 ±0.15	98.88 ±0.10	99.12 ±0.03	99.18 ±0.02	99.15 ±0.02
Multiclass								
	Number of PCA components							
Dataset size (%)	N=2	N=3	N=4	N=5	N=6	N=7	N=8	N=9
100	85.05±1.40	85.41±1.19	85.98±1.04	85.54±1.07	86.59±0.92	86.73 ±1.08	84.61±1.08	84.16±1.30
75	86.97 ±0.89	85.98 ±0.75	85.86 ±0.75	86.11 ±1.29	86.01 ±0.95	86.10 ±0.78	86.22 ±0.79	85.87 ±0.79
50	88.15 ±0.94	87.79 ±0.99	87.30 ±0.56	87.38 ±0.76	87.90 ±0.75	88.69 ±1.28	88.19 ±0.85	88.26 ±0.82
25	87.22 ±1.78	85.42 ±1.77	86.46 ±1.95	86.11 ±1.74	87.18 ±1.35	87.70 ±1.26	85.54 ±1.14	85.02 ±1.23

Table 5. Performance obtained for different kernels in the three binary classification scenarios.

Controls *vs* Pneumonia			
Linear	Polynomial (d=2)	Polynomial (d=3)	RBF
97.74 ±0.33	97.59 ±0.24	96.88 ±0.87	95.67 ±1.04
Bacterial *vs* Viral pneumonia			
84.88 ±1.13	86.57 ±0.81	84.56 ±1.34	82.34 ±1.73
Viral *vs* COVID19 pneumonia			
99.36 ±0.01	99.24 ±0.02	99.18 ±0.01	98.25 ±0.02

the class that yields a minimum reconstruction error. This alternative, applied in combination with ensemble classification, has shown a high performance in previous works.[128–130] However, it is difficult to use an ensemble version when input images are not analyzed as a whole but divided into patches. With reference to our application context, patterns associated with COVID-19 can be distributed in different locations of the image. According to the severity of the infection, they can be widespread or bounded in small regions. This last situation can be highly problematic when trying to automatize the diagnosis for one main reason. It is possible that most of the regions within the ensemble are labeled as 'controls' because they are not affected by the pulmonary affection, whereas only a small number of regions are identified as 'covid patient'. In this case, combining the results from individual patches is not straightforward. Employing majority voting is not an optimum solution, especially when a non-severe affection is present. Previous studies have weighted the contribution of individual patches according to a specific residual e.g. uncertainty in Bayesian frameworks.[109] There are some scenarios in which two lung regions are labeled with opposite diagnoses and both classifier's decisions are correct, especially if the pneumonia is not widespread. In order to overcome this issue, features extracted from individual parts of the images are treated as a whole in the classification stage to optimize the diagnostic process.

Another remarkable aspect of the proposed method is the high performance obtained without requiring a previous preprocessing of the images. The use of artificial intelligence for the automatic detection of different pathologies is widespread, e.g. neurological disorders such as Parkinson's[16,47,131] or Alzheimer's.[13,132,133] When analyzing patterns associated with brain anatomy or function, most of these techniques require a spatial correspondence between the images of all subjects. This can be obtained by employing operations based on spatial transformations such as registration or normalization. However, the application of these approaches to CXR images is much harder for several reasons. First, there is a high variability in the size and shape of lungs. And most important, there are discrepancies in the position of each patient inside the scanner for all the images acquired. When trying to apply spatial transformations to mitigate these issues, it is possible to introduce high levels of noise that invalidate the results obtained. We have developed an accurate tool that does not require any additional preprocessing to get a high performance. In fact, the information extracted from the sparse coding methodology in addition to the computation of the reconstruction errors perform consistently well despite no spatial correspondence between the different images is computed.

It is worth mentioning that our method can be an excellent option in contexts where the applicability of deep learning approaches is not straightforward. Alternatives based on deep learning have shown an ideal solution when applied to medical imaging in a wide range of scenarios. Therefore, previous works have demonstrated a high performance when used to detect pneumonia.[107,134–136] The main issue is that this kind of techniques require a high number of training samples in order to learn the features that allows the detection of a specific pathology. The implementation of a global repository of COVID-19 images would address this problem. However, collaboration between different medical centers is not always possible. For this reason, it is important to note that the design of our method mitigates the influence of the data size in performance. Another crucial difference between our proposal and deep learning methods is related to the computational burden. Specifically, the number of mathematical operations performed by our approach is considerably lower than the ones employed in deep learning. In fact, the computational time of the proposed method was a 75% lower than the time needed for the deep learning approach employed in Ref.,[109] where the same dataset was used. This time was computed as the average between the computational time obtained by the different number of components used. This evidences the suitability of our framework in research centres with reduced computational resources. Moreover, the high performance obtained in the multiclass classification shows that the tool proposed in this work can be successfully employed in a real scenario. These results reveal the usefulness of this technique not only for identifying pneumonia, but to properly identify the cause of this pathology.

7. Conclusions and Future Work

In this paper, we propose a method to process images and extract informative features using sparse coding. This is addressed by dividing the images into different patches, and storing them into a matrix. PCA is applied to this matrix in order to build a dictionary that maximizes the differences in the representation of the different patches. Once a sparse representation is obtained, the reconstruction errors are computed and entered as features into a classifier. We obtained a 86.73% of accuracy and AUC of 0.87 in the multiclass scenario, where different types of pneumonia are distinguished. These results validate the applicability of the method as an aid for clinicians in a real and complex context. Besides, the reduced computational cost compared to deep learning while preserving a large performance paves the way to use this methodology with other image modalities such as MRI or PET. Future versions of the method could employ more sophisticated algorithms such as Enhanced Probabilistic Neural Network,[137] Neural Dynamic Classification,[138] Dynamic Ensemble Learning Algorithm[139] or Finite Element Machine[140] in order to improve the classification performance.

8. Funding

This work was supported by the MCIN/ AEI/10.13039/501100011033/ and FEDER "Una manera de hacer Europa" under the RTI2018-098913-B100 project, by the Consejería de Economía, Innovación, Ciencia y Empleo (Junta de Andalucía) and FEDER under CV20-45250, A-TIC-080-UGR18, B-TIC-586-UGR20 and P20-00525

projects.

Bibliography

1. Y. Zheng and X. Hu, "Concurrent prediction of finger forces based on source separation and classification of neuron discharge information," *International Journal of Neural Systems*, vol. 31, no. 06, p. 2150010, 2021.
2. W. Y. Peh, J. Thomas, E. Bagheri, R. Chaudhari, S. Karia, R. Rathakrishnan, V. Saini, N. Shah, R. Srivastava, Y.-L. Tan, and J. Dauwels, "Multicenter validation study of automated classification of pathological slowing in adult scalp electroencephalograms via frequency features," *International Journal of Neural Systems*, vol. 31, no. 06, p. 2150016, 2021.
3. Y. Xue, H. Zhu, and F. Neri, "A self-adaptive multiobjective feature selection approach for classification problems," *Integrated Computer-Aided Engineering*, 2021.
4. J. Buenaposada and L. Baumela, "Improving multiclass boosting-based object detection," *Integrated Computer-Aided Engineering*, vol. 28, pp. 1–16, 07 2020.
5. S. Liapis, K. Christantonis, V. Chazan-Pantzalis, A. Manos, D. Filippidou, and C. Tjortjis, "A methodology using classification for traffic prediction: Featuring the impact of COVID-19," *Integrated Computer-Aided Engineering*, vol. 28, pp. 1–19, 07 2021.
6. B. Mwangi, T. Tian, and J. Soares, "A review of feature reduction techniques in neuroimaging," *Neuroinformatics*, vol. 12, pp. 229–244, 2013.
7. J. Górriz, A. Lassl, J. Ramírez, D. Salas-Gonzalez, C. Puntonet, and E. Lang, "Automatic selection of rois in functional imaging using gaussian mixture models," *Neuroscience Letters*, vol. 460, pp. 108–111, 2009.
8. E. de la Hoz Franco, E. de la Hoz Correa, A. Ortiz, J. Lopera, and A. Martínez-Álvarez, "Feature selection by multi-objective optimisation: Application to network anomaly detection by hierarchical self-organising maps," *Knowl. Based Syst.*, vol. 71, pp. 322–338, 2014.
9. Y. Xue, Y. Tang, X. Xu, J. Liang, and F. Neri, "Multi-objective feature selection with missing data in classification," *IEEE Transactions on Emerging Topics in Computational Intelligence*, pp. 1–10, 2021.
10. U. Acharya, V. K. Sudarshan, H. Adeli, J. Santhosh, J. E. W. Koh, S. D. Puthankatti, and A. Adeli, "A novel depression diagnosis index using nonlinear features in EEG signals," *European Neurology*, vol. 74, pp. 79 – 83, 2015.
11. A. Ortiz, J. M. Górriz, J. Ramírez, and F. Martínez-Murcia, "Automatic ROI selection in structural brain MRI using som 3D projection," *PloS one*, vol. 9, p. e93851, 04 2014.
12. H. Adeli, "Inventing the future of neurology: Integrated wavelet-chaos-neural network models for knowledge discovery and automated EEG-based diagnosis of neurological disorders," in *2008 IEEE International Conference on Information Reuse and Integration*, 2008, pp. x–xi.
13. J. E. Arco, J. Ramírez, C. G. Puntonet, J. M. Górriz, and M. Ruz, "Improving short-term prediction from MCI to AD by applying Searchlight analysis," in *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*, 2016, pp. 10–13.
14. J. E. Arco, P. Díaz-Gutiérrez, J. Ramírez, and M. Ruz, "Atlas-based classification algorithms for identification of informative brain regions in fMRI," *Neuroinformatics*, vol. 18, pp. 219–236, 2019.
15. N. Mammone, L. Bonanno, S. De Salvo, A. Bramanti, P. Bramanti, H. Adeli, C. Ieracitano, M. Campolo, and F. C. Morabito, "Hierarchical clustering of the electroencephalogram spectral coherence to study the changes in brain connectivity in Alzheimer's disease," in *2016 IEEE Congress on Evolutionary Computation (CEC)*, 2016, pp. 1241–1248.
16. A. Ortiz, J. Munilla, F. J. Martínez-Murcia, J. M. Górriz, and J. Ramírez, "Empirical functional PCA for 3D image feature extraction through fractal sampling," *International Journal of Neural Systems*, vol. 29, no. 02, p. 1850040, 2019.
17. K. S. Omar, P. Mondal, N. S. Khan, M. R. K. Rizvi, and M. N. Islam, "A machine learning approach to predict autism spectrum disorder," in *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, 2019, pp. 1–6.
18. H. S. Nogay and H. Adeli, "Machine learning (ML) for the diagnosis of autism spectrum disorder (ASD) using brain imaging:," *Reviews in the Neurosciences*, vol. 31, no. 8, pp. 825–841, 2020. [Online]. Available: https://doi.org/10.1515/revneuro-2020-0043
19. J. M. Górriz, J. Ramírez, F. Segovia, F. J. Martínez, M.-C. Lai, M. V. Lombardo, S. Baron-Cohen, , and J. Suckling, "A machine learning approach to reveal the neurophenotypes of autisms," *International Journal of Neural Systems*, vol. 29, no. 07, p. 1850058, 2019.
20. O. K. Cura and A. Akan, "Classification of epileptic EEG signals using synchrosqueezing transform and machine learning," *International Journal of Neural Systems*, vol. 31, no. 05, p. 2150005, 2021.
21. Y. Varatharajah, R. K. Iyer, B. M. Berry, G. A. Worrell, and B. H. Brinkmann, "Seizure forecasting and the preictal state in canine epilepsy," *International Journal of Neural Systems*, vol. 27, no. 01, p. 1650046, 2017.
22. A. Shoeb and J. Guttag, "Application of machine learning to epileptic seizure detection," in *Proceed-*

ings of the 27th International Conference on International Conference on Machine Learning, ser. ICML'10. Madison, WI, USA: Omnipress, 2010, p. 975–982.

23. K. Kourou, T. P. Exarchos, K. P. Exarchos, M. V. Karamouzis, and D. I. Fotiadis, "Machine learning applications in cancer prognosis and prediction," *Computational and Structural Biotechnology Journal*, vol. 13, pp. 8–17, 2015.
24. X. Yu, Q. Zhou, S. Wang, and Y.-D. Zhang, "A systematic survey of deep learning in breast cancer," *International Journal of Intelligent Systems*, vol. n/a, no. n/a, 2021.
25. M. Amrane, S. Oukid, I. Gagaoua, and T. Ensarİ, "Breast cancer classification using machine learning," in *2018 Electric Electronics, Computer Science, Biomedical Engineerings' Meeting (EBBT)*, 2018, pp. 1–4.
26. A. Lozano, J. S. Suárez, C. Soto-Sánchez, F. J. Garrigós, J. J. Martínez-Álvarez, J. M. Ferrández, and E. Fernández, "Neurolight: A deep learning neural interface for cortical visual prostheses," *International journal of neural systems*, p. 2050045, 2020.
27. P. Lara-Benítez, M. Carranza-García, and J. C. Riquelme, "An experimental review on deep learning architectures for time series forecasting," *International Journal of Neural Systems*, vol. 31, no. 03, p. 2130001, 2021. [Online]. Available: http://dx.doi.org/10.1142/S0129065721300011
28. M. A. Ozdemir, O. K. Cura, and A. Akan, "Epileptic EEG classification by using time-frequency images for deep learning," *International Journal of Neural Systems*, vol. 31, no. 08, p. 2150026, 2021.
29. P. Lara-Benítez, M. Carranza-García, J. García-Gutiérrez, and J. Riquelme, "Asynchronous dual-pipeline deep learning framework for online data stream classification," *Integrated Computer-Aided Engineering*, vol. 27, pp. 1–19, 2020.
30. J. García-González, J. Ortiz-De-Lazcano-Lobato, R. M. Luque Baena, and E. López-Rubio, "Background subtraction by probabilistic modeling of patch features learned by deep autoencoders," *Integrated Computer-Aided Engineering*, vol. 27, pp. 1–13, 03 2020.
31. S. Hamreras, B. Boucheham, M. A. Molina-Cabello, R. Benítez-Rochel, and E. López-Rubio, "Content based image retrieval by ensembles of deep learning object classifiers," *Integrated Computer-Aided Engineering*, vol. 27, pp. 1–15, 04 2020.
32. Y. Hou, T. Chen, X. Lun, and F. Wang, "A novel method for classification of multi-class motor imagery tasks based on feature fusion," *Neuroscience Research*, 2021.
33. H. Noğay and H. Adeli, "Detection of epileptic seizure using pretrained deep convolutional neural network and transfer learning," *European Neurology*, vol. 83, pp. 602–614, 01 2020.
34. J. Benito-Picazo, E. Domínguez, E. J. Palomo, and E. López-Rubio, "Deep learning-based video surveillance system managed by low cost hardware and panoramic cameras," *Integr. Comput. Aided Eng.*, vol. 27, pp. 373–387, 2020.
35. M. Wiering, M. Schutten, A. Millea, A. Meijster, and L. Schomaker, "Deep support vector machines for regression problems," in *International Workshop on Advances in Regularization, Optimization, Kernel Methods, and Support Vector Machines: theory and applications*, 2013, pp. 53–54.
36. J. Gasienica-Józkowy, M. Knapik, and B. Cyganek, "An ensemble deep learning method with optimized weights for drone-based water rescue and surveillance," *Integr. Comput. Aided Eng.*, vol. 28, pp. 221–235, 2021.
37. M. J. Gómez-Silva, A. de la Escalera, and J. M. Armingol, "Back-propagation of the mahalanobis istance through a deep triplet learning model for person re-identification," *Integr. Comput. Aided Eng.*, vol. 28, pp. 277–294, 2021.
38. A. Valikhani, A. Jaberi Jahromi, S. Pouyanfar, I. M. Mantawy, and A. Azizinamini, "Machine learning and image processing approaches for estimating concrete surface roughness using basic cameras," *Computer-Aided Civil and Infrastructure Engineering*, vol. 36, no. 2, pp. 213–226, 2021.
39. J. A. Park, C. M. Yeum, and T. D. Hrynyk, "Learning-based image scale estimation using surface textures for quantitative visual inspection of regions-of-interest," *Computer-Aided Civil and Infrastructure Engineering*, vol. 36, no. 2, pp. 227–241, 2021.
40. U. R. Acharya, S. L. Oh, Y. Hagiwara, J. H. Tan, H. Adeli, and D. P. Subha, "Automated EEG-based screening of depression using deep convolutional neural network," *Computer Methods and Programs in Biomedicine*, vol. 161, pp. 103–113, 2018.
41. G. B. a. Martins, J. a. P. Papa, and H. Adeli, "Deep learning techniques for recommender systems based on collaborative filtering," *Expert Systems*, vol. 37, no. 6, p. e12647, 2020.
42. Y. Xue, P. Jiang, F. Neri, and J. Liang, "A multi-objective evolutionary approach based on graph-in-graph for neural architecture search of convolutional neural networks," *International Journal of Neural Systems*, vol. 31, no. 09, p. 2150035, 2021.
43. W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: A comprehensive review," *Neural Computation*, vol. 29, pp. 2352–2449, 2017.
44. A. Ortiz, J. Munilla, J. M. Górriz, and J. Ramírez, "Ensembles of deep learning architectures for the early diagnosis of the Alzheimer's disease," *International Journal of Neural Systems*, vol. 26, no. 07, p. 1650025, 2016.
45. H. Nogay and H. Adeli, "Detection of epileptic seizure using pretrained deep convolutional neural network and transfer learning," *European Neurol-*

ogy, vol. 83, pp. 602 – 614, 2020.
46. F. J. Martínez-Murcia, A. Ortiz, J. Górriz, J. Ramírez, and D. Castillo-Barnes, "Studying the manifold structure of Alzheimer's disease: A deep learning approach using convolutional autoencoders," *IEEE Journal of Biomedical and Health Informatics*, vol. 24, pp. 17–26, 2020.
47. F. Martínez-Murcia, J. M. Gorriz, J. Ramírez, and A. Ortiz, "Convolutional neural networks for neuroimaging in Parkinson's disease: Is preprocessing needed?" *International Journal of Neural Systems*, vol. 28, 07 2018.
48. M. Leming, J. M. Górriz, and J. Suckling, "Ensemble deep learning on large, mixed-site fMRI datasets in autism and other tasks," *International journal of neural systems*, p. 2050012, 2020.
49. A. H. Ansari, P. J. Cherian, A. Caicedo, G. Naulaers, M. De Vos, and S. Van Huffel, "Neonatal seizure detection using deep convolutional neural networks," *International Journal of Neural Systems*, vol. 29, no. 04, p. 1850011, 2019.
50. K. Thurnhofer-Hemsi, E. López-Rubio, N. Roé-Vellvé, and M. A. Molina-Cabello, "Multiobjective optimization of deep neural networks with combinations of lp-norm cost functions for 3D medical image super-resolution," *Integr. Comput. Aided Eng.*, vol. 27, no. 3, pp. 233–251, 2020.
51. D. Castillo-Barnes, F. J. Martínez-Murcia, A. Ortiz, D. Salas-Gonzalez, J. Ramírez, and J. M. Górriz, "Morphological characterization of functional brain imaging by isosurface analysis in Parkinson's disease," *International journal of neural systems*, p. 2050044, 2020.
52. B. Li, H. Peng, X. Luo, J. Wang, X. Song, M. J. Pérez-Jiménez, and A. Riscos-Núñez, "Medical image fusion method based on coupled neural P systems in nonsubsampled shearlet transform domain," *International Journal of Neural Systems*, vol. 31, no. 01, p. 2050050, 2021.
53. J. Zhu, C. Tan, J. Yang, G. Yang, and P. Lio', "Arbitrary scale super-resolution for medical images," *International Journal of Neural Systems*, vol. 31, no. 10, p. 2150037, 2021.
54. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.
55. M. L. D. Dias, A. N. Maia, A. R. da Rocha Neto, and J. a. P. P. Gomes, "Parsimonious minimal learning machine via multiresponse sparse regression," *International Journal of Neural Systems*, vol. 30, no. 05, p. 2050023, 2020.
56. S. Arabi, A. Haghighat, and A. Sharma, "A deep-learning-based computer vision solution for construction vehicle detection," *Computer-Aided Civil and Infrastructure Engineering*, vol. 35, no. 7, pp. 753–767, 2020.
57. J. Kim, H. Kim, and T. Hong, "Automated classification of indoor environmental quality control using stacked ensembles based on electroencephalograms," *Computer-Aided Civil and Infrastructure Engineering*, vol. 35, no. 5, pp. 448–464, 2020.
58. Q. Quan, F. He, and L. Haoran, "A multi-phase blending method with incremental intensity for training detection networks," *The Visual Computer*, vol. 37, pp. 245–259, 02 2021.
59. P. Vuttipittayamongkol and E. Elyan, "Improved overlap-based undersampling for imbalanced dataset classification with application to epilepsy and Parkinson's disease," *International Journal of Neural Systems*, vol. 30, no. 08, p. 2050043, 2020.
60. A. Ortiz, F. J. Martinez-Murcia, J. L. Luque, A. Giménez, R. Morales-Ortega, and J. Ortega, "Dyslexia diagnosis by EEG temporal and spectral descriptors: An anomaly detection approach," *International Journal of Neural Systems*, vol. 30, no. 07, p. 2050029, 2020.
61. T. Zhou, D. Tao, and X. Wu, "Manifold elastic net: a unified framework for sparse dimension reduction," *Data Mining and Knowledge Discovery*, vol. 22, pp. 340–371, 2010.
62. L. He, H. Chen, and L. Carin, "Tree-structured compressive sensing with variational Bayesian analysis," *IEEE Signal Processing Letters*, vol. 17, pp. 233–236, 2010.
63. J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2009.
64. W. Dong, L. Zhang, and G. Shi, "Centralized sparse representation for image restoration," in *2011 International Conference on Computer Vision*, 2011, pp. 1259–1266.
65. Y. Zheng and J. C. Gee, "Estimation of image bias field with sparsity constraints," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 255–262.
66. K. Jia, X. Wang, and X. Tang, "Optical flow estimation using learned sparse model," in *2011 International Conference on Computer Vision*, 2011, pp. 2391–2398.
67. V. Maihami and F. Yaghmaee, "A review on the application of structured sparse representation at image annotation," *Artificial Intelligence Review*, vol. 48, pp. 331–348, 2016.
68. R. Rigamonti, M. A. Brown, and V. Lepetit, "Are sparse representations really relevant for image classification?" *CVPR 2011*, pp. 1545–1552, 2011.
69. D. Dai and W. Yang, "Satellite image classification via two-layer sparse coding with biased image representation," *IEEE Geoscience and Remote Sensing Letters*, vol. 8, no. 1, pp. 173–176, 2011.
70. A. Soltani-Farani and H. R. Rabiee, "When pixels team up: Spatially weighted sparse coding for hy-

perspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 1, pp. 107–111, 2015.
71. C. Bao, L. He, and Y. Wang, "Linear spatial pyramid matching using non-convex and non-negative sparse coding for image classification," 2015.
72. Y. Shi, Y. Wan, K. Wu, and X. Chen, "Non-negativity and locality constrained laplacian sparse coding for image classification," *Expert Systems with Applications*, vol. 72, pp. 121–129, 2017.
73. J. J. Thiagarajan, K. N. Ramamurthy, D. Rajan, A. Spanias, A. Puri, and D. Frakes, "Kernel sparse models for automated tumor segmentation," *International Journal on Artificial Intelligence Tools*, vol. 23, no. 03, p. 1460004, 2014.
74. J. Han, H. Chang, L. Loss, K. Zhang, F. L. Baehner, J. W. Gray, P. Spellman, and B. Parvin, "Comparison of sparse coding and kernel methods for histopathological classification of gliobastoma multiforme," in *2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 2011, pp. 711–714.
75. N. Weiss, D. Rueckert, and A. Rao, "Multiple sclerosis lesion segmentation using dictionary learning and sparse coding," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2013*, K. Mori, I. Sakuma, Y. Sato, C. Barillot, and N. Navab, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 735–742.
76. J. Cheng, F. Yin, D. W. K. Wong, D. Tao, and J. Liu, "Sparse dissimilarity-constrained coding for glaucoma screening," *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 5, pp. 1395–1403, 2015.
77. Y. Xu, X. Gao, S. Lin, D. Wong, J. Liu, D. Xu, C.-Y. Cheng, C. Cheung, and T. Wong, "Automatic grading of nuclear cataracts from slit-lamp lens images using group sparsity regression," *Medical image computing and computer-assisted intervention : MICCAI ... International Conference on Medical Image Computing and Computer-Assisted Intervention*, vol. 16 Pt 2, pp. 468–75, 2013.
78. N. Nayak, H. Chang, A. Borowsky, P. Spellman, and B. Parvin, "Classification of tumor histopathology via sparse feature learning," in *2013 IEEE 10th International Symposium on Biomedical Imaging*, 2013, pp. 410–413.
79. H. Chang, N. M. Nayak, P. Spellman, and B. Parvin, "Characterization of tissue histopathology via predictive sparse decomposition and spatial pyramid matching," *Medical image computing and computer-assisted intervention : MICCAI ... International Conference on Medical Image Computing and Computer-Assisted Intervention*, vol. 16 Pt 2, pp. 91–8, 2013.
80. L. Deng and D. Yu, "Deep learning: Methods and applications," *Foundations and Trends® in Signal Processing*, vol. 7, no. 3–4, pp. 197–387, 2014.
81. J. E. Arco, C. González-García, P. Díaz-Gutiérrez, J. Ramírez, and M. Ruz, "Influence of activation pattern estimates and statistical significance tests in fMRI decoding analysis," *Journal of Neuroscience Methods*, vol. 308, pp. 248–260, 2018.
82. K. Roy, D. Banik, D. Bhattacharjee, and M. Nasipuri, "Patch-based system for classification of breast histology images using deep learning," *Computerized Medical Imaging and Graphics*, vol. 71, pp. 90–103, 2019.
83. W. E. Vinje and J. L. Gallant, "Sparse coding and decorrelation in primary visual cortex during natural vision," *Science*, vol. 287, no. 5456, pp. 1273–1276, 2000.
84. J. J. Hunt, P. Dayan, and G. J. Goodhill, "Sparse coding can predict primary visual cortex receptive field changes induced by abnormal visual input," *PLOS Computational Biology*, vol. 9, no. 5, pp. 1–17, 2013.
85. M. Beyeler, E. L. Rounds, K. D. Carlson, N. Dutt, and J. L. Krichmar, "Neural correlates of sparse coding and dimensionality reduction," *PLOS Computational Biology*, vol. 15, no. 6, pp. 1–33, 2019.
86. J. Li, W. Zhou, S. Yuan, Y. Zhang, C. Li, and Q. Wu, "An improved sparse representation over learned dictionary method for seizure detection," *International Journal of Neural Systems*, vol. 26, no. 01, p. 1550035, 2016.
87. I. T. Jolliffe, *Principal Component Analysis and Factor Analysis*. Springer New York, 1986, pp. 115–128.
88. L. Khedher, J. Ramírez, J. M. Górriz, A. Brahim, and F. Segovia, "Early diagnosis of Alzheimer's disease based on partial least squares, principal component analysis and support vector machine using segmented MRI images," *Neurocomputing*, vol. 151, pp. 139 – 150, 2015.
89. M. López, J. Ramírez, J. M. Górriz, D. Salas-Gonzalez, I. Álvarez, F. Segovia, and C. Puntonet, "Automatic tool for Alzheimer's disease diagnosis using PCA and Bayesian classification rules," *Electronics Letters*, vol. 45, pp. 389–391, 2009.
90. M. López, J. Ramírez, J. M. Górriz, I. Illan, D. Salas-Gonzalez, F. Segovia, and R. Chaves, "SVM-based CAD system for early detection of the Alzheimer's disease using kernel PCA and LDA," *Neuroscience letters*, vol. 464, pp. 233–8, 09 2009.
91. B. Schölkopf, A. Smola, and K. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, 1998.
92. J. HUO, Y. GAO, W. YANG, and H. YIN, "Multi-instance dictionary learning for detecting abnormal events in surveillance videos," *International Journal of Neural Systems*, vol. 24, no. 03, p. 1430010, 2014.
93. A. Apicella, F. Isgrò, R. Prevete, and G. Tamburrini, "Middle-level features for the explanation of

classification systems by sparse dictionary methods," *International Journal of Neural Systems*, vol. 30, no. 08, p. 2050040, 2020.

94. Q. YUAN, W. ZHOU, S. YUAN, X. LI, J. WANG, and G. JIA, "Epileptic EEG classification based on kernel sparse representation," *International Journal of Neural Systems*, vol. 24, no. 04, p. 1450015, 2014.
95. M. Liu, D. Zhang, and D. Shen, "Ensemble sparse classification of Alzheimer's disease," *NeuroImage*, vol. 60, no. 2, pp. 1106–1116, 2012.
96. J. E. Arco, J. Ramírez, J. M. Górriz, and M. Ruz, "Data fusion based on Searchlight analysis for the prediction of Alzheimer's disease," *Expert Systems with Applications*, vol. 185, p. 115549, 2021.
97. B. Boser, I. Guyon, and V. Vapnik, "A training algorithm for optimal margin classifier," *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*, vol. 5, 08 1996.
98. L. Breiman, "Machine learning, volume 45, number 1 - springerlink," *Machine Learning*, vol. 45, pp. 5–32, 10 2001.
99. B.-S. Yang, X. Di, and T. Han, "Random forests classifier for machine fault diagnosis," *Journal of Mechanical Science and Technology*, vol. 22, pp. 1716–1725, 09 2008.
100. Kaggle. (2020) Chest X-Ray Images (Pneumonia) dataset. [Online]. Available: https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia?
101. D. S. Kermany, M. Goldbaum, W. Cai, C. C. Valentim, H. Liang, S. L. Baxter, A. McKeown, G. Yang, X. Wu, F. Yan, J. Dong, M. K. Prasadha, J. Pei, M. Y. Ting, J. Zhu, C. Li, S. Hewett, J. Dong, I. Ziyar, A. Shi, R. Zhang, L. Zheng, R. Hou, W. Shi, X. Fu, Y. Duan, V. A. Huu, C. Wen, E. D. Zhang, C. L. Zhang, O. Li, X. Wang, M. A. Singer, X. Sun, J. Xu, A. Tafreshi, M. A. Lewis, H. Xia, and K. Zhang, "Identifying medical diagnoses and treatable diseases by image-based deep learning," *Cell*, vol. 172, no. 5, pp. 1122 – 1131.e9, 2018.
102. Kaggle. (2020) Chest X-Ray Images (COVID-19 & Pneumonia). [Online]. Available: https://www.kaggle.com/prashant268/chest-xray-covid19-pneumonia
103. R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*, ser. IJCAI'95. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1995, p. 1137–1143.
104. J. N. Mandrekar, "Receiver operating characteristic curve in diagnostic test assessment," *Journal of Thoracic Oncology*, vol. 5, no. 9, pp. 1315 – 1316, 2010.
105. K. Hajian-Tilaki, "Receiver operating characteristic (ROC) curve analysis for medical diagnostic test evaluation," *Caspian journal of internal medicine*, vol. 4, pp. 627–635, 09 2013.
106. E. L. Allwein, R. E. Schapire, and Y. Singer, "Reducing multiclass to binary: A unifying approach for margin classifiers," *J. Mach. Learn. Res.*, vol. 1, p. 113–141, 2001.
107. R. Alizadehsani, D. Sharifrazi, N. H. Izadi, J. H. Joloudari, A. Shoeibi, J. M. Górriz, S. Hussain, J. E. Arco, Z. A. Sani, F. Khozeimeh, A. Khosravi, S. Nahavandi, S. M. S. Islam, and U. R. Acharya, "Uncertainty-aware semi-supervised method using large unlabelled and limited labeled COVID-19 data," 2021.
108. L. Li, L. Qin, Z. Xu, Y. Yin, X. Wang, B. Kong, J. Bai, Y. Lu, Z. Fang, Q. Song, K. Cao, D. Liu, G. Wang, Q. Xu, X. Fang, S. Zhang, J. Xia, and J. Xia, "Artificial intelligence distinguishes COVID-19 from community acquired pneumonia on chest ct," *Radiology*, vol. 296, p. 200905, 03 2020.
109. J. E. Arco, A. Ortiz, J. Ramirez, F. J. Martinez-Murcia, Y.-D. Zhang, and J. M. Górriz, "Uncertainty-driven ensembles of deep architectures for multiclass classification. application to COVID-19 diagnosis in chest X-ray images," 2020.
110. J. E. Arco, A. Ortiz, J. Ramírez, F. J. Martínez-Murcia, Y.-D. Zhang, J. Broncano, M. Álvaro Berbís, J. R. del Val, A. Luna, and J. M. Górriz, "Probabilistic combination of eigenlungs-based classifiers for COVID-19 diagnosis in chest CT images," 2021.
111. C. Li, D. Dong, L. Li, W. Gong, X. Li, Y. Bai, M. Wang, Z. Hu, Y. Zha, and J. Tian, "Classification of severe and critical COVID-19 using deep learning and radiomics," *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 12, pp. 3585–3594, 2020.
112. K. Panetta, F. M. Sanghavi, S. Agaian, and N. Madan, "Automated detection of COVID-19 cases on radiographs using shape-dependent fibonacci-p patterns," *IEEE Journal of Biomedical and Health Informatics*, vol. 25, pp. 1852–1863, 2021.
113. M. Frid-Adar, R. Amer, O. Gozes, J. Nassar, and H. Greenspan, "COVID-19 in CXR: From detection and severity scoring to patient disease monitoring," *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 6, pp. 1892–1903, 2021.
114. A. Narin, C. Kaya, and Z. Pamuk, "Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks," *Pattern Analysis and Applications*, pp. 1 – 14, 2021.
115. M. E. H. Chowdhury, T. Rahman, A. Khandakar, R. Mazhar, M. A. Kadir, Z. B. Mahbub, K. R. Islam, M. S. Khan, A. Iqbal, N. A. Emadi, M. B. I. Reaz, and M. T. Islam, "Can AI help in screening viral and COVID-19 pneumonia?" *IEEE Access*, vol. 8, pp. 132 665–132 676, 2020.
116. A. Makris, I. Kontopoulos, and K. Tserpes, "COVID-19 detection from chest X-ray images us-

ing deep learning and convolutional neural networks," in *11th Hellenic Conference on Artificial Intelligence*. Association for Computing Machinery, 2020, p. 60–66.

117. H. Ko, H. Chung, W. S. Kang, K. W. Kim, Y. Shin, S. J. Kang, J. H. Lee, Y. J. Kim, N. Y. Kim, H. Jung, and J. Lee, "COVID-19 pneumonia diagnosis using a simple 2D deep learning framework with a single chest CT image: Model development and validation," *J Med Internet Res*, vol. 22, no. 6, p. e19569, 2020.
118. S. Tabik, A. Gómez-Ríos, J. L. Martín-Rodríguez, I. Sevillano-García, M. Rey-Area, D. Charte, E. Guirado, J. L. Suárez, J. Luengo, M. A. Valero-González, P. García-Villanova, E. Olmedo-Sánchez, and F. Herrera, "COVIDGR dataset and COVID-SDNet methodology for predicting COVID-19 based on chest X-ray images," *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 12, pp. 3595–3605, 2020.
119. M. Ghaderzadeh, F. Asadi, R. Jafari, D. Bashash, H. Abolghasemi, and M. Aria, "Deep convolutional neural network–based computer-aided detection system for COVID-19 using multiple lung scans: Design and implementation study," *J Med Internet Res*, vol. 23, no. 4, p. e27468, 2021.
120. T. Ozturk, M. Talo, E. A. Yildirim, U. B. Baloglu, O. Yildirim, and U. Rajendra Acharya, "Automated detection of COVID-19 cases using deep neural networks with X-ray images," *Computers in Biology and Medicine*, vol. 121, p. 103792, 2020.
121. N. K. Mishra, P. Singh, and S. D. Joshi, "Automated detection of COVID-19 from CT scan using convolutional neural network," *Biocybernetics and Biomedical Engineering*, vol. 41, no. 2, pp. 572–588, 2021.
122. S. H. Kassania, P. H. Kassanib, M. J. Wesolowskic, K. A. Schneidera, and R. Detersa, "Automatic detection of coronavirus disease (COVID-19) in X-ray and CT images: A machine learning based approach," *Biocybernetics and Biomedical Engineering*, vol. 41, no. 3, pp. 867–879, 2021.
123. M. R. Ibrahim, S. Youssef, and K. M. Fathalla, "Abnormality detection and intelligent severity assessment of human chest computed tomography scans using deep learning: a case study on SARS-COV-2 assessment," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1 – 24, 2021.
124. S. Wang, D. Nayak, D. Guttery, X. Zhang, and Y. Zhang, "COVID-19 classification by fgcnet with deep feature fusion from graph convolutional network and convolutional neural network," *Information Fusion*, vol. 68, 2021.
125. E. Jangam, A. A. D. Barreto, and C. S. R. Annavarapu, "Automatic detection of COVID-19 from chest CT scan and chest X-rays images using deep learning, transfer learning and stacking," *Applied Intelligence*, p. 1—17, June 2021.
126. M. Rahimzadeh, A. Attar, and S. M. Sakhaei, "A fully automated deep learning-based network for detecting COVID-19 from a new and large lung CT scan dataset," *Biomedical Signal Processing and Control*, vol. 68, p. 102588, 2021.
127. J. Ortega, J. Asensio-Cubero, J. Gan, and A. Ortiz, "Classification of motor imagery tasks for BCI with multiresolution analysis and multiobjective feature selection," *BioMedical Engineering OnLine*, vol. 15, 07 2016.
128. S. Shekhar, V. M. Patel, and R. Chellappa, "Analysis sparse coding models for image-based classification," in *2014 IEEE International Conference on Image Processing (ICIP)*, 2014, pp. 5207–5211.
129. J. Xu, L. Ding, and S. Sun, "Supervised Bayesian sparse coding for classification," in *2014 International Joint Conference on Neural Networks (IJCNN)*, 2014, pp. 319–326.
130. J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1794–1801.
131. T. J. Hirschauer, H. Adeli, and J. A. Buford, "Computer-aided diagnosis of Parkinson's disease using enhanced probabilistic neural network," *Journal of Medical System*, vol. 39, no. 11, 2015.
132. D. Castillo-Barnes, J. Ramírez, F. Segovia, F. J. Martínez-Murcia, D. Salas-Gonzalez, and J. M. Górriz, "Robust ensemble classification methodology for i123-ioflupane SPECT images and multiple heterogeneous biomarkers in the diagnosis of Parkinson's disease," *Frontiers in Neuroinformatics*, vol. 12, p. 53, 2018.
133. J. Górriz, J. Ramírez, A. Ortíz, F. Martínez-Murcia, F. Segovia, J. Suckling, M. Leming, Y. Zhang, J. Álvarez Sánchez, G. Bologna, P. Bonomini, F. Casado, D. Charte, F. Charte, R. Contreras, A. Cuesta-Infante, R. Duro, A. Fernández-Caballero, and J. Fernández, "Artificial intelligence within the interplay between natural and artificial computation: Advances in data science, trends and applications," *Neurocomputing*, vol. 410, pp. 237–270, 2020. [Online]. Available: http://dx.doi.org/10.1016/j.neucom.2020.05.078
134. D. S. Kermany, M. Goldbaum, W. Cai, C. C. Valentim, H. Liang, S. L. Baxter, A. McKeown, G. Yang, X. Wu, F. Yan, J. Dong, M. K. Prasadha, J. Pei, M. Y. Ting, J. Zhu, C. Li, S. Hewett, J. Dong, I. Ziyar, A. Shi, R. Zhang, L. Zheng, R. Hou, W. Shi, X. Fu, Y. Duan, V. A. Huu, C. Wen, E. D. Zhang, C. L. Zhang, O. Li, X. Wang, M. A. Singer, X. Sun, J. Xu, A. Tafreshi, M. A. Lewis, H. Xia, and K. Zhang, "Identifying medical diagnoses and treatable diseases by image-based deep learning," *Cell*, vol. 172, no. 5, pp. 1122–1131.e9, 2018.
135. A. Mittal, D. Kumar, M. Mittal, T. Saba, I. Abunadi, A. Rehman, and s. Roy., "Detecting

pneumonia using convolutions and dynamic capsule routing for chest X-ray image," *Sensors*, vol. 20, no. 4, p. 1068, 2020.

136. Z. Wang, Y. Xiao, Y. Li, J. Zhang, F. Lu, M. Hou, and X. Liu, "Automatically discriminating and localizing COVID-19 from community-acquired pneumonia on chest X-rays," *Pattern Recognition*, vol. 110, p. 107613, 2021.
137. M. Ahmadlou and H. Adeli, "Enhanced probabilistic neural network with local decision circles: A robust classifier," *Integr. Comput. Aided Eng.*, vol. 17, pp. 197–210, 2010.
138. M. H. Rafiei and H. Adeli, "A new neural dynamic classification algorithm," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 12, pp. 3074–3083, 2017.
139. K. Alam, N. Siddique, and H. Adeli, "A dynamic ensemble learning algorithm for neural networks," *Neural Computing and Applications*, vol. 32, no. 12, p. 8675–8690, 2020.
140. D. R. Pereira, M. A. Piteri, A. N. D. Souza, J. P. Papa, and H. Adeli, "FEMa: a finite element machine for fast learning," *Neural Computing and Applications*, vol. 32, pp. 6393–6404, 2019.