Pattern Recognition 120 (2021) 108135

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Deep neural networks ensemble to detect COVID-19 from CT scans

Lerina Aversano[a], Mario Luca Bernardi[a,*], Marta Cimitile[b], Riccardo Pecori[a]

[a] *University of Sannio, Benevento (BN), Italy*
[b] *Unitelma Sapienza University, Rome, Italy*

ARTICLE INFO

Article history:
Received 28 December 2020
Revised 17 June 2021
Accepted 27 June 2021
Available online 7 July 2021

Keywords:
Deep learning
CT Scan images
COVID-19
Coronavirus

ABSTRACT

Research on Coronavirus Disease 2019 (COVID-19) detection methods has increased in the last months as more accurate automated toolkits are required. Recent studies show that CT scan images contain useful information to detect the COVID-19 disease. However, the scarcity of large and well balanced datasets limits the possibility of using detection approaches in real diagnostic contexts as they are unable to generalize. Indeed, the performance of these models quickly becomes inadequate when applied to samples captured in different contexts (e.g., different equipment or populations) from those used in the training phase. In this paper, a novel ensemble-based approach for more accurate COVID-19 disease detection using CT scan images is proposed. This work exploits transfer learning using pre-trained deep networks (e.g., VGG, Xception, and ResNet) evolved with a genetic algorithm, combined into an ensemble architecture for the classification of clustered images of lung lobes. The study is validated on a new dataset obtained as an integration of existing ones. The results of the experimental evaluation show that the ensemble classifier ensures effective performance, also exhibiting better generalization capabilities.

1. Introduction

COVID-19, which appeared for the first time in China in December 2019, rapidly spread around the world and became a pandemic. It has caused a devastating effect on both the public health and the global economy, changing our daily lives. The rapid spread of the COVID-19 epidemic constitutes a relevant research challenge, from different points of view: humanitarian as well as technological. In particular, COVID-19 is the first pandemic in the digital era, therefore there is a large amount of publicly available information collected by various institutions allowing the enrollment of the entire scientific community to provide their contribution to analyze such a complex and multifaceted context.

Research communities of different research centers are actively participating in developing effective diagnostic mechanisms and solutions for its treatment.

In the medical field, the use of medical images is widespread. Medical professionals and radiologists commonly use medical images for diagnosing and prescribing treatment of diseases. Historically, template-based and retrieval-based approaches have been proposed and are now superseded by methods based on deep neural networks. The network architecture used for these pattern recognition tasks may include encoder-decoder frameworks, fully connected networks, convolutional neural networks (CNN) that extracts the visual features from images and are trained on the dataset. Specifically, the research studies in the field of medicine or biotechnology aimed at mitigating this pandemic greatly exploit recent developments in convolutional neural networks especially for pattern recognition tasks [1].

The major challenges in the COVID-19 rapid detection are related to the long duration of the tests for the diagnosis of the disease, and the long time required to provide physical equipment for tests. This leads to the lack of an adequate number of kits for COVID-19 detection available worldwide.

According to this, experts evaluate the adoption of AI-driven tools [2] for the collection of multiple data types and the detection of anomaly patterns due to COVID-19. Among these useful data, CT (Computed Tomography) scan images represent an alternative diagnosis method with several advantages of leveraging them [3].

In this direction, Deep learning (DL) algorithms could provide a useful tool for the COVID-19 disease detection. Indeed, Deep learning techniques have been successfully applied in many similar problems such as skin cancer classification [4], Parkinson and brain disease classification [5], and pneumonia detection through chest X-ray images [6]. As a consequence, several AI models have been proposed for the automatic diagnosis of COVID-19 from medical images [6,7]. Several models exploit images of the chest, obtained through CT scans that provide a 3D view of organs and a convenient way to analyze the disease effects on the impacted locations.

* Corresponding author.
E-mail addresses: aversano@unisannio.it (L. Aversano), bernardi@unisannio.it (M.L. Bernardi), marta.cimitile@unitelmasapienza.it (M. Cimitile), rpecori@unisannio.it (R. Pecori).

https://doi.org/10.1016/j.patcog.2021.108135

Some recent studies highlight that the sensitivity of CT for COVID-19 infection is 98% compared to the RT-PCR (Reverse Transcriptase-Polymerase Chain Reaction) sensitivity of 71% [8,9]. However, a limit of these applications is the scarcity of large heterogeneous and well-balanced datasets. This reduces the possibility of using detection approaches in real diagnostic contexts as they are unable to generalize to wider populations, demographics, or geographies [10,11]. Moreover, the performance of these models seems inadequate when applied to samples captured in different contexts (e.g., different equipment or populations) compared to those used for the neural networks training phase [11].

In this study, a novel DL-based ensemble approach for automatic and accurate COVID-19 disease detection using CT scan images is proposed. The ensemble classifier is obtained as a combination of three pre-trained deep neural networks (i.e., VGG, Xception, and ResNet) evolved with a direct coding scheme based on genetic programming. In particular, three single classifiers are built to analyze three different groups of clustered images, one for each lung lobe. This allows one to specialize each single classifier on a specific lung area to increase the classification performance. To validate this approach, this study introduces a balanced dataset obtained as an integration of two existing ones. The goal is to overcome the limitations of similar studies by evaluating, in a real scenario, if the proposed approach improves the performance of the classification and exhibits better generalization capabilities across different datasets.

To this aim, the validation reported in this study compares the ensemble approach with the pre-trained models carefully analyzing their performance. Specifically, the described assessment quantifies the impact of dataset integration, lobe-driven CT images clustering, and the ensemble architecture on the final end-to-end detection performance.

The remainder of the paper is organized as follows. In Section 2, a background on convolutional neural networks and genetic algorithms is provided. Section 3 presents and discusses the most relevant related work, highlighting differences and common aspects. The proposed approach is described in Section 4, whereas the experiment description is explained in Section 5. An in-depth discussion of the experiment results is reported in Section 6. Finally, Section 7 highlights some threats to the validity of the described experiments, while Section 8 discusses some final remarks and future research directions.

2. Background

2.1. Convolutional neural networks

In this study, we compare the performance obtained by using three state-of-the-art alternative pre-trained convolutional neural networks (CNNs) [12], i.e., ResNet50, VGG19, and Xception. The reason for the pre-training is that the refinement of a pre-trained network with transfer learning is generally much faster and easier than training from scratch, by requiring a minimal amount of data and computing resources. Transfer learning uses knowledge of one type of problem to solve similar problems and allows the usage of a pre-trained network in order to learn a new activity.

VGG19 [13], shown in Fig. 1, was conceived and created by the Visual Geometry Group of the University of Oxford for the ImageNet Large Scale Visual Recognition Challenge (ILSVRC-2014). This type of CNN receives as input a fixed-size RGB image (224 x 224), which is passed through a stack of convolutional layers using 3 x 3 filters, with a step size of 1 pixel, covering the whole notion of an image. It usually has 16 convolutional layers followed by 3 fully connected layers. The first two have 4096 channels each, the third one contains 1000 channels and performs ILSVRC classification. The final layer is characterized by a softmax activation function. All hidden layers are equipped with the ReLU function.

ResNet-50 [14] is a very deep convolutional neural network, made up of 50 layers, which, with the help of a technique known as “skip connection”, paved the way for the so-called residual networks. Moreover, ResNet-50 neural networks are an innovative solution to the problem of the escaping gradient. As shown in Fig. 2, a ResNet-50 stacks the levels and initially skips some of them in the first phases of the training, reusing the activation functions from previous levels. The jump initially compresses the network into only a few levels, which allows for faster learning. Then, when the network trains again, all the layers are expanded and the “residual” parts of the network explore more and more the feature space of the source image.

The Xception model [15] usually overtakes the previously described CNNs in both speed and accuracy. It relies on two main points: “Depth-wise Separable Convolution” and “Shortcuts between Convolutional blocks” as in the ResNet. As shown in Fig. 3, the architecture of the Xception model is based on depth-wise separable convolutional layers and consists of three major sections: Entry Flow, Middle Flow, and Exit Flow. Each Convolution and Separable Convolution layer is followed by a batch normalization layer. The Xception model takes the principles of Inception to an extreme, instead of partitioning input data into several compressed chunks, it maps the spatial correlations for each output channel separately and then performs a 1×1 depth-wise convolution to capture cross-channel correlation.

2.2. Evolutionary algorithms for neural networks optimization

In this study, the design of the single classifiers is driven by an evolutionary algorithm [16], that is responsible to search for effective adaptations of pre-trained deep neural networks to the given classification task. Specifically, a generic genetic algorithm is used to optimize both the neural network architectures fine-tuning and the related hyper-parameters. Genetic algorithms are search-based algorithms inspired by the process of natural selection and genetics [17]. They start by constructing an initial set of candidate solutions (population size) and calculate their fitness function in order to evaluate each chromosome in the population. After the fitness value is computed, some genetic operations are performed to select and evolve the population according to the typical genetic process, characterized by bio-inspired operators such as selection, mutation, and crossover. The selection process consists of preserving strong subjects and eliminating the weak ones. In the mutation process, new subjects are produced by randomly combining the existing ones. The crossover operator simulates the reproduction and biological crossover process, and the new subjects are produced by using the genetic material of the existing subjects (parents). The evolutionary process ends when the desired accuracy or the maximal generation number is reached. The genetic algorithms are recently broadly used in several application domains [17] since, given their capability of global search, they are considered as a novel and essential approach related to the modern intelligent calculation. However, the advantages of genetic algorithms are several: (i) they do not require additional information that generally is not available in real-world problems, (ii) they perform better with respect to traditional methods, (iii) they are suitable when the search space is very large and a large number of parameters is involved, like in the case of deep neural networks.

3. Related work

Machine Learning (ML) and DL techniques have been used in medical domains, obtaining good classification results [18,19]. More specifically, ML and DL classifiers have been used to extract the

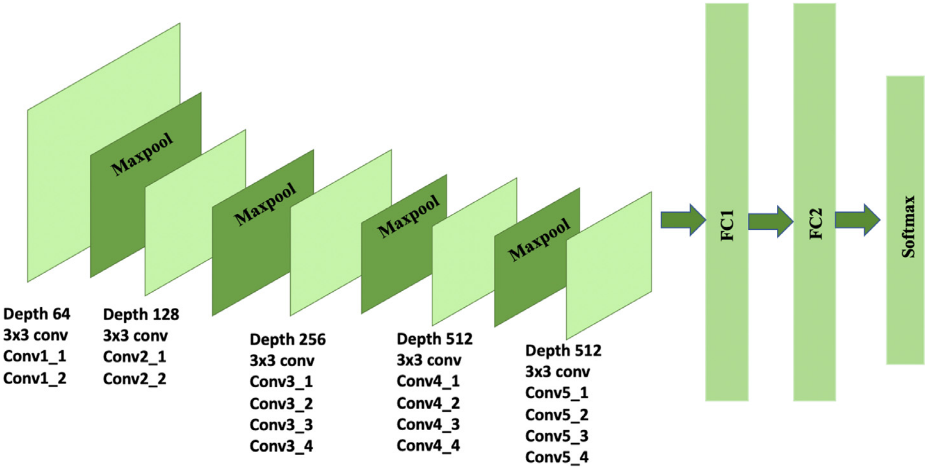

Fig. 1. VGG convolutional neural network.

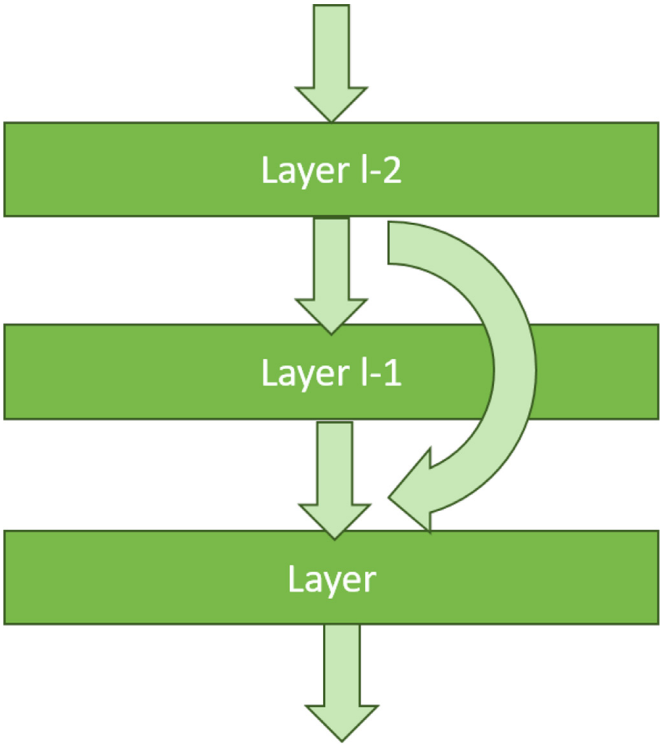

Fig. 2. ResNet network model.

relevant features from medical images to perform a disease diagnosis and/or prediction [20]. Basing on these premises, in the last months, some research introduced the application of DL for the COVID-19 diagnosis [21]. Some studies [22] propose deep neural networks for either CT scans and Chest X-rays (CXRs) to detect COVID-19 positive cases. Here, we focus on the adoption of DL on CT scan datasets to perform COVID-19 diagnosis. For example, in [9] a weakly supervised deep learning framework for automatic detection and classification of COVID-19 infected regions is introduced. It uses retrospectively extracted CT images from multi-scanners and multi-centres obtaining good results on typical examples of images of COVID-19 and other pneumonia cases. Similarly, authors in [23] adopt various deep CNN based approaches for detecting COVID-19 patients from chest CT images. The F1-score obtained on a dataset[1] composed of images collected from a scientific paper preprint is 0.867. A 3D deep learning model, referred to as COVNet, for detecting COVID-19 patients is also proposed in [24]. On the adopted dataset (composed of 4356 CT scans of 3322 patients) the AUC value for COVID-19 is 0.96. In [25] a combination of DL and Q-deformed entropy handcrafted features have been used for discriminating patients with COVID-19, pneumonia, and healthy cases by their CT lung scans. The best performance for the proposed LSTM network on the adopted dataset (it is obtained by using CT images of COVID-19 patients extracted by Radiopaedia[2]) is 99.68%. Authors in [26] test 10 convolutional neural networks to discriminate COVID-19 from non-COVID-19 cases: AlexNet, VGG-16, VGG-19, SqueezeNet, GoogleNet, MobileNet-V2, ResNet-18, ResNet-50, ResNet-101, and Xception. Among all networks, the best performance on the ad hoc built dataset (it is not available online) is obtained by ResNet-101 with an AUC of 0.994. The study proposed by Mei et al. [27] uses a deep CNN for the training step and both a support vector machine (SVM) and a random forest and multilayer perceptron (MLP) classifiers to detect patients with COVID-19 according to clinical information. Authors create a dataset (not available online) of chest CT scans from 905 patients obtaining in the best case (MLP) similar performance with respect to the classification of a senior radiologist. As also highlighted in [8,11], the bottleneck of these studies is the limited number of high quality publicly available comprehensive datasets. According to this, in [8] the authors adopt transfer learning along with data augmentation to detect COVID-19 patients from a small dataset. This work has considered ResNet18, ResNet50, ResNet101, and SqueezeNet architectures for the experimental evaluation. The best results are obtained by the ResNet18 pre-trained transfer learning-based model (validation accuracy = 97.32% and testing accuracy = 99.4%). This issue is also discussed in [11], where a cross dataset is obtained by integrating CT images of COVID-19 patients in order to obtain a more realistic scenario where images come from different sources. However, these images are also extracted from pre-prints of scientific articles, and data augmentation techniques are applied to increase the number of data. The adopted deep learning models drop from 87.68% to 56.16% of accuracy highlighting the necessity of further studies.

In this paper, differently from the aforementioned studies, we propose a new dataset obtained by integrating publicly available datasets. This allowed us to consider images coming from differ-

[1] https://arxiv.org/abs/2003.13865

[2] www.radiopaedia.org

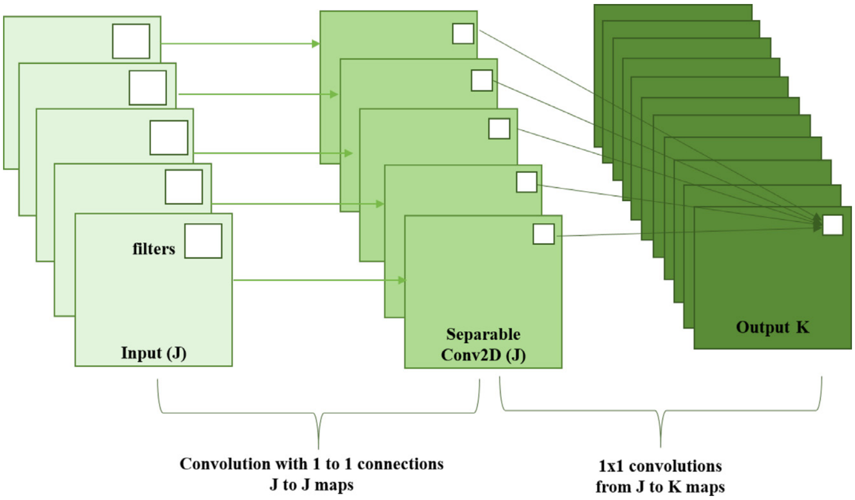

Fig. 3. Xception convolutional neural network.

ent sources avoiding the need of synthetic data. The adoption of the integrated dataset also allowed us to increase the generalization capabilities of the proposed approach with respect to the existing ones. However, differently from other approaches, herein we adopt an ensemble-based classification specialized on different lung lobes performing clustering of CT scan sequences. As proposed in [28], this allows each neural classifier to specialize in a well-defined lobe, capturing its specific patterns of damages and their distribution. The literature about ensemble learning with DL for COVID-19 detection from CT scans is discussed in [29]. Here the authors propose an ensemble-based approach to improve performance with respect to a single classifier. Another deep-LSTM ensemble approach is also proposed in [30]. Similarly, in [31], the authors use an ensemble-based approach to reduce the high degree of inter-observer variability in determining COVID-19. However, we differ from these approaches because our ensemble classifier relies on a preliminary image clustering activity. This clustering activity is aimed to group similar images (one group for each lobe) allowing the neural network to focus on disease patterns specific to a certain lobe. Moreover, another novelty introduced in this study is the adoption of a genetic algorithm, consisting of an evolutionary process to discover the best architecture adaptation of the pre-trained models to adapt them to the COVID-19 detection task. Indeed, we evaluate that this algorithm can be suitable in the proposed domain given the large search space and the high number of parameters.

4. Approach

The classification methodology adopted in this paper is based on a hierarchical multiple classifier schema using majority voting, as depicted in Fig. 4.

Specifically, the ensemble architecture is made of two essential components:

- multiple deep neural networks (i.e., referred to as single classifiers) based on pre-trained models that are adapted and re-trained for the COVID-19 detection task by means of an evolutionary algorithm;
- a voting strategy, used to take decisions based on the outcomes of the single classifiers.

The upper part of Fig. 4 depicts the ensemble training process: it starts with merging, clustering, and partitioning procedures used to generate the input for the adopted neural network models during the re-training phase. The output of the clustering step is the set of three sub-datasets (D_I, D_S, D_M) each one used to train a single classifier (C_I, C_S, C_M).

The lower part of Fig. 4 depicts the end-to-end classification schema of the ensemble, revealing also its internal structure. The ensemble is built by combining the re-trained single classifiers (C_I, C_S, C_M) with a majority voting strategy. During the inference phase, for unlabeled and clustered instances of CT scans, the single classifiers are applied to produce the input for the voter (i.e., the super classifier). The voting is performed by using a classic majority voting approach.

In the following subsections, the dataset clustering, the single classifiers optimized by means of an evolutionary algorithm, and the resulting ensemble will be further described.

4.1. Dataset clustering

In this phase, the CT scan images, coming from the two considered datasets, are clustered into three sub-datasets. In particular, referring to the right lung segments (Superior lobe, Inferior lobe, and Middle lobe) we named these three image clusters as follows: $\mathrm{D_S}$, $\mathrm{D_M}$, $\mathrm{D_I}$. Figure 5 reports the three groups of CT scan images belonging to the three lung segments.

The adopted clustering procedure starts with an encoding step. In this step the network pre-trained on the image dataset has been used to extract numerical feature descriptors of each image [32]. The images are submitted to the input layer of the network. The outputs coming out of one or more intermediate layers in the network can be used as the feature representation of the image. These features are used to perform a subsequent clustering step with the goal of grouping images that are similar with respect to a distance metric. The selected distance metric is a variant of the Structural Similarity Index Metric (SSIM) that exploits separate functional measures for luminance, contrast, and structural similarity between two images. In order to perform the calculation of local sample statistics, overlapping windows are used, wherein the functional measures are weighted by means, for instance, of a Gaussian-like profile. The overall index is a combination of the three components (structural, luminance, and contrast correlations) yielding to a general form where the three parameters are used to mediate the relative importance of the components themselves. From the original definition of the SSIM, several variants have been proposed by adding multi-scale support and additional components to make the index more robust to alterations that do not change semantic (scaling, rotations and so on). In this work, the four-component multi-scale structural

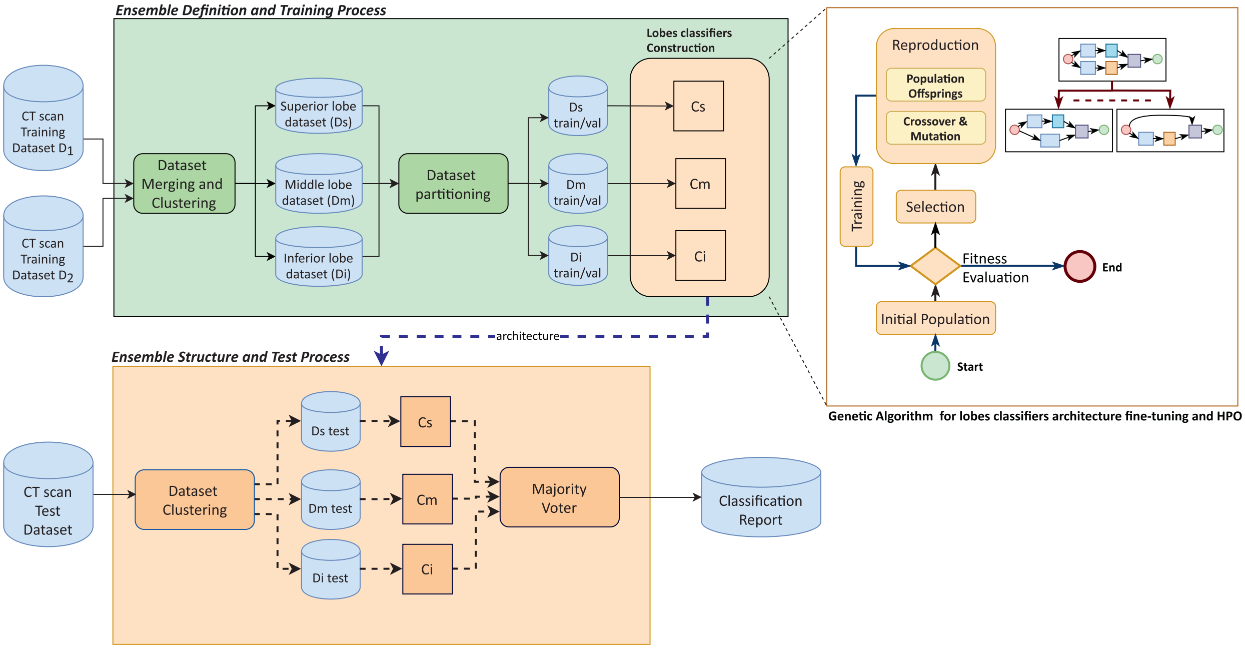
Ensemble Definition and Training Process
CT scan Training Dataset D_1
CT scan Training Dataset D_2
Dataset Merging and Clustering
Superior lobe dataset (Ds)
Middle lobe dataset (Dm)
Inferior lobe dataset (Di)
Dataset partitioning
Ds train/val
Dm train/val
Di train/val
Lobes classifiers Construction
Cs
Cm
Ci
architecture
Reproduction
Population Offsprings
Crossover & Mutation
Training
Selection
Fitness Evaluation
End
Initial Population
Start
Genetic Algorithm for lobes classifiers architecture fine-tuning and HPO
Ensemble Structure and Test Process
CT scan Test Dataset
Dataset Clustering
Ds test
Dm test
Di test
Cs
Cm
Ci
Majority Voter
Classification Report

Fig. 4. Overview of the proposed hierarchical multiple classifiers approach.

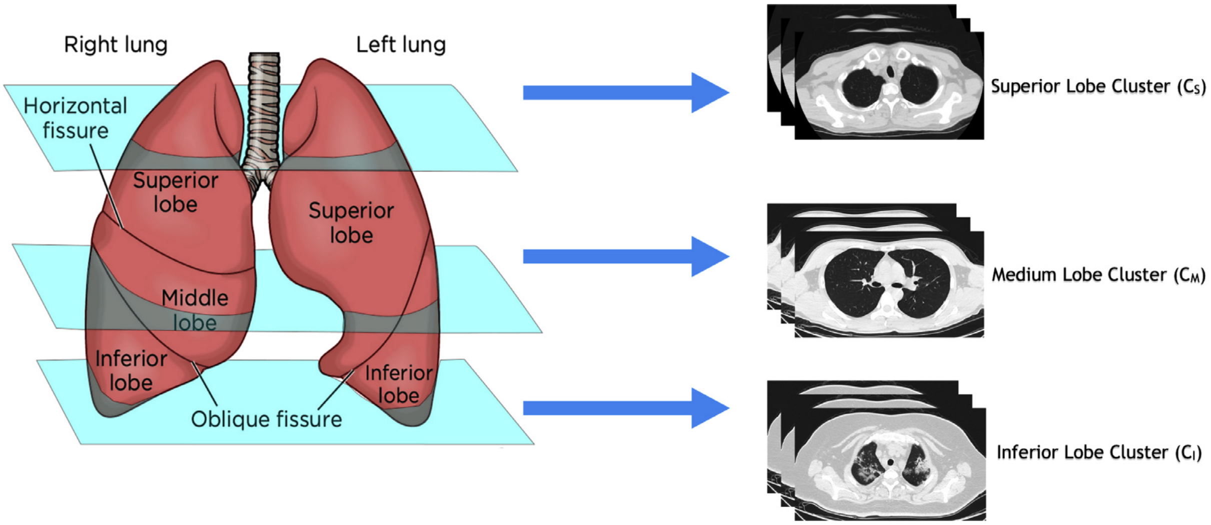

Fig. 5. Clusters of CT scans images for lung segments using K-means with k=3.

similarity index defined in [33] is used, since it is particularly suitable for radiological images (e.g., CT scans) as highlighted in [34].

Clustering is performed by using the K-means algorithm [35], which works by partitioning n observations into k mutually exclusive clusters and placing each observation into the cluster with the nearest mean. Each cluster in the partition is defined by a centroid, or centre, i.e., the point to which the sum of distances from all objects in that cluster is minimized, so that also the total intra-group variance is minimized. Differently from other approaches, K-means requires choosing the number of clusters. The suitable number of clusters is computed, in this paper, with an automatic method based on silhouette coefficients [36]. A silhouette value $s(i)$ is a measure, for each image i, of how similar that image is to images in its own cluster compared to images in other clusters. Hence the silhouette of a cluster is a plot (silhouette plot) of $s(i)$, ranked in decreasing order, for all images in the cluster. Global measures of the silhouettes can be given averaging them over the entire dataset, i.e., the mean of all individual silhouettes, also referred to as the average silhouette width for the dataset. The silhouette coefficients can be then used as a criterion to decide the best number of clusters [37]: the process applies k-means for k varying between two and $n-1$, where n is the number of images, choosing the value of k for which the average silhouette width for the entire dataset is maximized.

As Fig. 6 shows, the optimal choice for the number of clusters is equal to three when choosing 4-MS-SSIM as image similarity metric. This confirms that, when choosing $k=3$, the CT scans are partitioned into three clusters that roughly belong to the major lungs segments (Fig. 5). From the classification point of view, it is important since grouped images of the same lobe tend to be similar making much easier for the neural network to focus on disease patterns specific to the lobe.

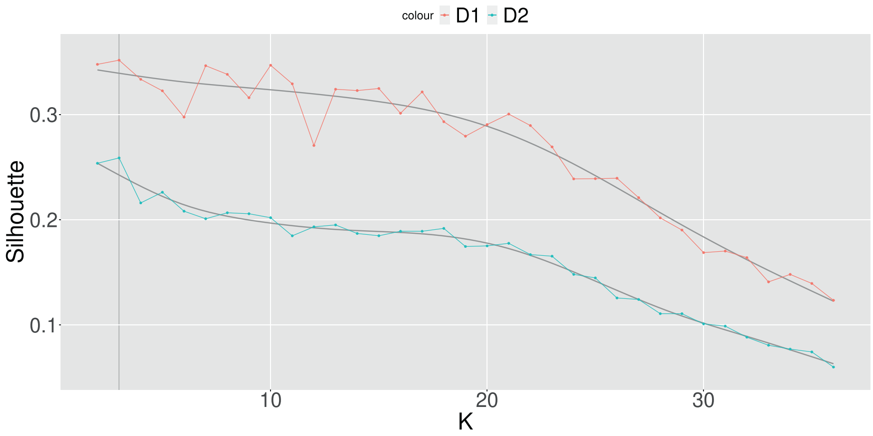

Fig. 6. Silhouette trend over the number of clusters for the merged datasets D_1 and D_2. The best value is obtained for k=3.

4.2. Single classifiers

As already pointed out, the design of the single classifiers exploits transfer learning and is based on an evolutionary algorithm. In particular, a genetic algorithm coding scheme is used for the representation of the pre-trained network models along with the optimization of the hyper-parameters in the training step.

As a consequence the single classifiers were built through a two-fold process procedure:

- generation of the training and validation sets;
- models re-training driven by the evolutionary algorithm and selection of the best performing classifiers.

Therefore, the classifiers were obtained using transfer learning-based models, given that they are recognized to be suitable for the medical image classification tasks [4]. In this work, transfer learning is used for the binary classification of chest CT scan images into two categories: COVID-19 and non-COVID-19. To this aim, three different pre-trained transfer-learning-based CNN models were used, namely, VGG19, ResNet, and Xception. Then the pre-trained transfer learning CNN models were re-trained for the binary classification of chest CT images into the aforementioned two classes. For each sub-dataset, the clustered set of images of the patients is composed of instances associated with a label (`Covid` or `Non-Covid`) and used for the training steps.

The overall process is highlighted in the upper part of Fig. 4: the first sub-process is depicted on the left side and shows dataset partitioning for the generation of the training/validation datasets. On the right side of the figure, the overall structure of the genetic algorithm (GA) used for AutoML is represented. Specifically, the algorithm executes an evolutionary process to discover the best architecture adaptation of the pre-trained models to classify the clustered CT scans. To this aim, it takes as input: (i) the set of predefined building blocks belonging to the pre-defined models (VGG, Xception, ResNet), (ii) the population size, (iii) the maximal generation number for the GA, and (iv) the image classification dataset. The starting population is initialized using random choices with a predefined population size and exploiting an encoding strategy able to represent a set of possible desired adaptations of the original model along with their hyper-parameters. Then, during evolution, the fitness function of each individual, which encodes a particular architecture of the pre-trained model, is evaluated on the input datasets. At this point, the parent individuals are selected based on the fitness function, and then generate a new offspring by applying suitable genetic operators (e.g., mutation and crossover). Finally, the population of individuals that survives into the next generation is selected by applying environmental selection to the current population, composed of the generated offspring population and the parent one. The evolution cycle proceeds until the optimal performance is obtained or the maximum number of iterations is reached.

More specifically, the procedure of the used GA can be detailed as follows:

1. Instantiate the initial population of individuals P (one for each pre-trained model), and train the CNNs represented by P using the validation accuracy as a fitness function;
2. Generate a set of λ offspring O, by applying the mutations to P;
3. Perform training on the λ modified pre-trained CNNs represented by offspring O, and assign the validation accuracy to the CNNs as a fitness function;
4. Select elite individuals from the union of the sets of P and O, and then replace P with them;
5. Repeat from step 2 until the stopping criterion is satisfied.

The algorithm starts from individuals based on the considered pre-trained models, giving to each model equal chances to produce individuals that perform well on the specific classification task. However, if a pre-trained model is not suitable and produces individuals that are less performing, it is quickly discarded since it will be not included in the elite set at step 3 after several iterations.

In order to represent a trainable model, starting from the original pre-trained model, the following encoding scheme was defined for both the hyper-parameters and the original model structure:

- **Final blocks training specifiers (FBTS)** - indicating which blocks of the original pre-trained model, starting from the last layers, have trainable weights and which ones are locked (specifically the string specifies 'L' for locked weights and 'T' when they are trainable);
- **number of FC layers** - indicating the number of layers of the fully connected (FC) final block used to re-train the solution on the given classification task;
- **FC neuron scheme** - specifying the size, in neurons, of each layer of the fully connected final block;
- **FC dropout scheme** - specifying where to insert dropout layers along with dropout rate probabilities (i.e., the probability of training a given node in a layer);
- **Optimizer** - specifying the optimization algorithm used to perform training.

4.3. Super classifier and ensemble learning

Ensemble learning [38] consists of a set of classifiers used to classify new instances in a combined fashion, i.e., the decision of each classifier is taken into account as a vote and all votes are combined together, according to a certain rule, producing a final overall classification decision as its output.

In the proposed approach, the ensemble employs the three single classifiers: C_I, C_S, C_M. The output of each classifier could be different because different training data are used in the single classifier itself. Moreover, we adopted the majority voting strategy for the combination of the outputs. Majority voting [39] counts the votes per label of all single classifiers and decides using the label with most votes. More precisely, for an unknown instance $\mathbf{x}$, each single basic classifier p produces a class-output probability function $f_{p,m}(\mathbf{x})$ for each m^{th} class. In general, with M classes (two in our case) and L classifiers (three in our case), the predicted class is the k^{th} class that collects the largest number of votes, given by the following formula:

$$\text{class}(\mathbf{x}) = \underset{k=1...M}{\text{argmax}}(\sum_{p}^{L}\{\text{if } k = \underset{m=1...M}{\text{argmax}}(f_{p,m}(\mathbf{x})) \text{ then } 1 \text{ else } 0\}), \tag{1}$$

where p iterates over the L classifiers and m and k over the M classes.

5. Experiment description

In the following sub-sections, the research questions, the dataset construction process and the evaluation setting are described.

5.1. Research questions

The research goals described in the introduction are detailed in the following research questions:

RQ1: *Is the performance of the best fine-tuned pre-trained deep neural network classifiers compatible with their usage in a real-world diagnostic context?*

This research question aims to assess, evaluate, and compare the performance of the analyzed transfer learning models to detect Covid-19 positive patients using CT scans to train the classifiers on different datasets. We have evaluated whether a model retrained on one dataset has good performance when tested on another dataset in order to be effective in a real context or not. The re-training process is driven by the evolutionary process described in the previous section and responsible for selecting the best performing network models through:

- architectural fine-tuning decisions, applied to the original model definitions (i.e., the layers of the original model that are locked and the ones to be retrained);
- hyper-parameters optimization (HPO) of the unlocked layers.

RQ2: *To what extent does lobe-driven clustering of CT scan images improve the performance of the pre-trained deep neural network classifiers?*

This research question aims at assessing and evaluating the performance of the best pre-trained deep neural network classifiers in detecting COVID-19 instances when CT scan images are clustered per lobe.

RQ3: *To what extent does the integration of the datasets improve the performance of the pre-trained deep neural network classifiers?*

This research question investigates whether the best networks trained on the integrated dataset provide better classification performance than the best networks trained on the single datasets.

RQ4: *Is the proposed ensemble model more effective than the pre-trained deep neural network classifiers across the different datasets?*

This research question investigates whether the ensemble classifier with majority voting improves the classification performance or not. In order to answer this question the performance of the ensemble classifier is compared with the performance of the single classifiers.

5.2. Datasets construction

The dataset construction represents a critical aspect for the diffusion and improvement of DL approaches for detecting COVID-19 patients exploiting CT images of the chest [8,11]. However, the publicly available datasets usually collect CT images having different formats, quality (several studies used images scanned by scientific papers or websites), and generated in a different way (i.e., real data and augmented data). Moreover, some studies [8] show that the performance of the existing approaches strongly depends on the adopted dataset, causing a reduced generalization of the obtained research results. These considerations motivated our idea to generate a new dataset as an integration of some existing ones.

The integration process consists of the following main steps:

1. selection of proper datasets;
2. cleaning and filtering of the selected datasets;
3. merging and balancing of the datasets.

The first step entails the selection of the datasets to integrate on the basis of their characteristics. According to this, a study of the publicly available existing datasets and a rigorous data acquisition process were performed. In particular, all the datasets containing artificially generated images were discarded, while we selected only datasets containing, for all the considered patients (those affected by COVID-19 and those not affected by the disease), the CT images for all the lung segments. Basing on the above criteria, two datasets, namely the Extensive COVID-19 X-Ray and CT Chest Images Dataset[3] and the Coronavirus (COVID-19) CC-19 dataset[4], were selected.

The first dataset contains both X-Ray and CT scan gray-scale images and its samples were increased through different augmentation techniques. The second dataset features only CT scan images coming from 89 subjects, 68 positive to COVI $D-$ 9 and 21 negative. The images are gray-scale and both 2D and 3D; moreover, they were collected in the earlier days of the epidemic from various hospitals in Chengdu, the capital city of Sichuan. For both datasets, we considered only CT scans acquired in the craniocaudal direction, with the patients lying in the supine position. Some examples of the considered images are reported in Fig. 5, wherein the inner dark cavities correspond to hypodense regions, i.e., the lungs, whereas the surrounding white hyperdense structures are soft tissues. The dark upper part of the images is the air, while the structures in the bottom part of the images represent the couch where the subject is lying down.

The pre-processing encompassed cleaning and filtering activities. Moreover, all the images having low quality were removed and all patients having a reduced or incomplete number of CT images (i.e., not all the lobes were shown) were removed, together with their corresponding CT scans.

The cleaned versions of the aforementioned datasets (hereinafter called D$_1$ and D$_2$) were then merged into an integrated dataset (D$_J$, joint dataset). To ensure that the integrated dataset was well balanced, a reduced number of patients from the larger dataset is used. As a matter of fact, D$_J$ contains 23,398 images, thereof 14,074 are labeled as COVID-19 and 9324 as non-COVID-19, respectively.

The overall dataset (the joint one and the two sub-datasets, D$_1$ and D$_2$) is freely available at this link[5]. The link contains also part

[3] https://doi.org/10.17632/8h65ywD2jr.3

[4] https://github.com/abdkhanstd/COVID-19

[5] https://bit.ly/34QJUSd

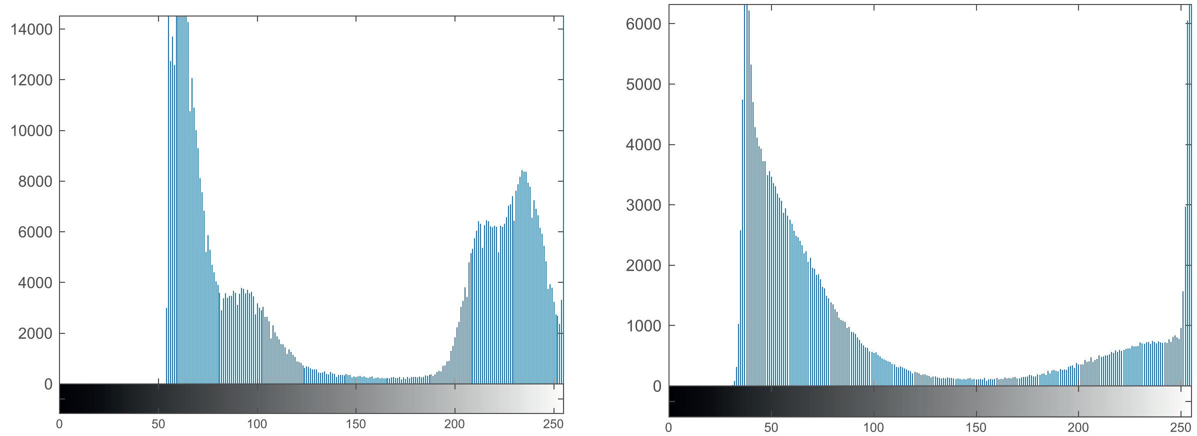

Fig. 7. Gray-scale profile of two images extracted from the two considered dataset (D_1 on the left and D_2 on the right) as regards COVID-19 patients and the medium lobes of a lung.

Table 1
Main characteristics of the considered datasets of CT scan images.

	Datasets			
Statistics	D_1	D_2	D_3	D_J
Number of images	14,312	9086	2481	23,398
Date of publication	2020	2020	2020	2021
COVID-19 images	7980	6094	1252	14,074
Non-COVID-19 images	6332	2992	1229	9324

of an additional dataset, namely SARS-COV-2 Ct-Scan Dataset[6], selected from the Web, which is used to test the performance on a dataset never used in the training phase and re-named, in its reduced form, as D_3. This dataset contains 1252 CT scans of patients that are positive for SARS-CoV-2 infection and 1229 for patients non-infected by SARS-CoV-2, collected from real patients in hospitals from São Paulo, Brazil, in 2020. Table 1 summarizes the main numerical characteristics of all considered datasets. For all the considered datasets the image format and resolution is the same, that is, jpeg format with a resolution of 512 × 512 pixels.

Finally, in Fig. 7 we show the gray-scale profile of two images extracted from D_1 (left) and D_2 (right) as regards the medium lobes of COVID-19 patients. The similarity of the profiles, after they were normalized, was measured through the correlation coefficient and the chi-square distance, which resulted to be 0.4822 (with perfect similarity equal to 1.0) and 11.37 (with exact similarity equal to 0.0), respectively.

5.3. Experimental setting

A set of experiments was performed to answer each research question introduced in Section 5.1.

Referring to RQ1, the performance of the VGG, Xception, and ResNet networks was evaluated by considering different combinations of the train and test sets: (D_1, D_1), (D_1, D_2), (D_2, D_1), and (D_2, D_2). The genetic algorithm, previously mentioned, was used for identifying the optimal pre-trained network models with the optimization of the hyper-parameters in the training step.

We tested different architectures of the basic component classifier considering the following ranges for the solution parameters:

- *Final blocks training specifiers* (FBTS): A three character string specifying an L or a T for each block to leave locked or to set as trainable;
- *FC number of layers*: the number of layers for the final fully connected block varies in the range [2,6];
- *FC neurons scheme*: the number of neurons for each layer, ranging in the interval [1, 256];
- *FC dropout scheme*: in this study it is changing in the interval [0:10; 0:25];
- *Optimizer*: the Stochastic Gradient Descent (SGD) with Nesterov's accelerated gradient [40], RMSProp, and Nadam optimization algorithms were considered.

RQ2 was explored by comparing the performance of the VGG, Xception, and ResNet networks trained on the clustered datasets with respect to their performance when they are trained on not clustered CT scans images.

In order to answer RQ3, the performance of VGG, Xception, and ResNet networks was evaluated and compared by considering different combinations of train and test sets: (D_1, D_1), (D_1, D_2), (D_2, D_1), (D_2, D_2), (D_J, D_1), (D_J, D_2), (D_1, D_J), (D_2, D_J).

Finally, for answering RQ4, the performance of the ensemble classifier for all the combinations of train and test datasets (as listed for RQ3) was compared with the one obtained to answer RQ3.

Moreover, as an additional experiment, the performance of the single and the ensemble classifiers was evaluated and compared when the testing is performed on an additional dataset (D_3), never used for the training.

As for the deep neural network classifiers, they have trained for a changing number of *layers* and a varying *number of epochs* with binary cross-entropy as a loss function. The deep neural network classifiers were implemented using Tensorflow[7], an open platform for machine learning tasks, and Keras[8], an open source neural network library written in Python. Similarly, the multiple ensemble classifier was developed using the Python programming language as well.

The metrics used to evaluate the training performance have been the *Accuracy* and the *Loss*. The loss function gives information on how well the dataset is modeled by the network. High values of loss mean that the predictions are totally wrong. On the other hand, if the loss is low, the prediction is performing well. The accuracy and loss function are inversely proportional: when accuracy is getting better, the loss is getting worse, and viceversa.

On the other hand, the classification results were evaluated using Accuracy, Precision (P), Recall (R), F1-score (F1), and ROC Area.

[6] https://www.kaggle.com/plameneduardo/sarscov2-ctscan-dataset

[7] https://www.tensorflow.org/

[8] https://keras.io/

Table 2
The best three results obtained by GA execution, trained and tested on the considered datasets.

Train	Base Model	FBTS	#FC Layers	FC Neuron Scheme	FC Dropout Scheme	Optimizer	Test	P	R	F1
D_1	VGG19	LLT	6	208 - 136 - 122 - 80 - 64 - 30	0.15 - N - 0.12 - N - 0.10 - N	NADAM	D_1	0.993	0.994	0.994
							D_2	0.691	0.683	0.687
							D_J	0.894	0.888	0.891
D_2	VGG19	LLT	6	258 - 156 - 112 - 92 - 54 - 32	0.15 - 0.15 - 0.10 - 0.10 - 0.10 - 0.10	NADAM	D_1	0.741	0.743	0.742
							D_2	0.996	0.993	0.995
							D_J	0.893	0.894	0.894
D_J	RESNET50	LLL	4	122 - 118 - 64 - 24	0.18 - N - 0.12 - 0.10	SGD	D_1	0.864	0.867	0.865
							D_2	0.895	0.893	0.894
							D_J	0.968	0.970	0.969
D_1	VGG19	TTT	5	202 - 106 - 88 - 48 - 28	0.25 - N - 0.15 - 0.15 - 0.10	NADAM	D_1	0.977	0.974	0.975
							D_2	0.626	0.618	0.622
							D_J	0.839	0.842	0.841
D_2	XCEPTION	LLT	5	214 - 104 - 55 - 55 - 32	0.25 - 0.10 - 0.14 - 0.13 - N	SGD	D_1	0.691	0.688	0.689
							D_2	0.972	0.974	0.973
							D_J	0.886	0.881	0.883
D_J	VGG19	LLT	6	188 - 162 - 142 - 117 - 60 - 32	0.15 - N - 0.12 - N - 0.10 - N	NADAM	D_1	0.864	0.865	0.864
							D_2	0.755	0.758	0.756
							D_J	0.965	0.964	0.964
D_1	RESNET50	LLL	4	112 - 118 - 84 - 31	0.18 - N - 0.12 - N	NADAM	D_1	0.962	0.959	0.961
							D_2	0.594	0.596	0.595
							D_J	0.720	0.720	0.720
D_2	RESNET50	LLL	4	94 - 48 - 44 - 32	0.25 - 0.10 - 0.10 - 0.10	RMSProp	D_1	0.594	0.595	05.95
							D_2	0.918	0.921	0.920
							D_J	0.770	0.763	0.766
D_J	VGG19	LLT	6	202 - 132 - 102 - 64 - 48 - 24	0.15 - N - 0.12 - N - 0.10 - N	RMSProp	D_1	0.836	0.831	0.833
							D_2	0.641	0.636	0.638
							D_J	0.951	0.952	0.952

They are computed as follows:

$$\text{Accuracy} = \frac{tp+tn}{tp+tn+fp+fn},$$

$$\text{P} = \frac{tp}{tp+fp},$$

$$\text{R} = \frac{tp}{tp+fn},$$

$$F1 = 2\frac{P \times R}{P+R},$$

where *tp* (true positives) is the number of correctly detected relevant instances, *fp* (false positive) is the number of irrelevant instances wrongly detected as relevant, *fn* (false negatives) is the number of relevant instances wrongly detected as irrelevant, *tn* (true negatives) is the number of irrelevant instances, correctly not detected.

Finally, the ROC Area (AUC) measures the probability that a relevant instance randomly selected is classified above a not relevant one.

The experiments were performed by using an Intel Core i9 7920X (18 cores), with 64GB of RAM and 2 GPUs (NVIDIA RTX 3090 24GB of RAM).

6. Results and discussion

In this section, we report the results obtained by performing the set of experiments described in subsection §5.3 that we discuss, in the following, according to the research questions defined in subsection §5.1.

As regards RQ1, Table 2 reports the performance (P, R, and F1) obtained by each pre-trained single neural network by using as train and test set all the possible combinations of D_1 and D_2 (reported in the columns titled “Train” and “Test”, respectively). For the sake of brevity, the table reports only the results corresponding to the best combination of the hyper-parameters (indicated in the columns from three to seven). From the table, we can observe that the best performance is obtained whenever the same dataset is used for training and testing. However, this condition is quite far from a real-world diagnostic context, wherein the test is performed on unknown new cases. A more realistic scenario is when different datasets are used for training and testing. In these conditions, we observe that the performance of the classifiers is, in some cases, quite reduced.

This is clear by looking at all the considered metrics in Table 2, in particular focusing on the F1-score. Considering, as an example, the networks trained on D_1, we can observe that when the test is performed on D_1 the best F1-score is obtained for VGG19 and its value is 0.994). Differently, when the test is performed on D_2, the best F1-score is obtained for VGG19 and its value is 0.687.

Similar considerations can be drawn when the training is performed on D_2.

Figure 8 confirms that, generally, the best average F1-score is obtained when training and testing are performed on the same dataset (a) with respect to the case where training and testing are performed on different datasets (b).

These outcomes corroborate the considerations discussed in [11], highlighting that the good performance obtained in several recent studies is strongly influenced by the use of the same dataset for training and testing the neural network.

In order to answer RQ2, some meaningful results are reported in Table 3. The table shows all the possible combinations of clustered datasets used for training and testing on images from the same lung lobe, as well as the relative performance obtained by the best considered neural network. Comparing Table 3 and Table 2, it emerges that the use of clustered datasets has a limited impact on the classification performance when the same dataset is used for training and testing.

However, by observing Fig. 9, it comes out that, using clustering, the F1-score of the single classifiers, trained and tested on the same dataset, is usually equal or lower compared with the best F1-score obtained using the corresponding non-clustered. Conversely, when the train and test datasets are different, the use of clustering

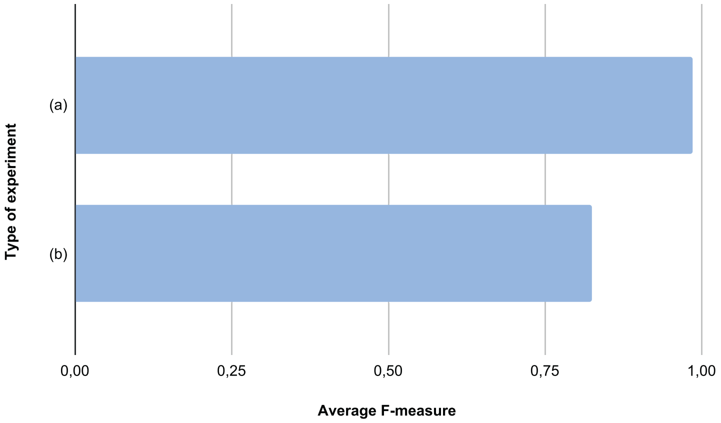

Fig. 8. Average F1-score performance comparison for the single classifiers: (a) training and testing on the same dataset, (b) training and testing on different datasets.

Table 3
The results of the best performing single neural networks trained and tested on datasets D_1, D_2, and D_J, clustered per lobe.

Train	Base Model	FBTS	FC #Layers	FC #Neurons Scheme	FC Dropouts Scheme	Optimizer	Test	P	R	F1
D_{1I}	VGG19	LLL	4	112 - 118 - 84 - 31	0.18 - N - 0.12 - N	SGD	D_1I	0.968	09.69	0.969
							D_{2I}	0.869	0.701	0.776
							D_{JI}	0.868	0.866	0.867
D_{1M}	XCEPTION	LLT	6	244 - 128 - 92 - 64 - 54 - 24	0.15 - 0.15 - 0.12 - 0.10 - 0.10 - 0.10	NADAM	D_{1M}	0.973	0.977	0.975
							D_{2M}	0.719	0.717	0.718
							D_{JM}	0.815	0.815	0.815
D_1S	XCEPTION	LTT	5	202 - 106 - 88 - 48 - 28	0.14 - 0.15 - 0.14 - 0.13 - 0.13	SGD	D_1S	0.956	0.956	0.956
							D_{2S}	0.720	0.715	0.718
							D_{JS}	0.864	0.864	0.864
D_{2I}	XCEPTION	LLT	5	214 - 104 - 55 - 55 - 32	0.25 - 0.10 - 0.14 - 0.13 - N	NADAM	D_{1I}	0.768	0.587	0.588
							D_{2I}	0.969	0.968	0.968
							D_{JI}	0.647	0.649	0.648
D_{2M}	VGG19	LLT	6	210 - 133 - 112 - 66 - 48 - 12	0.15 - N - 0.12 - N - 0.10 - N	NADAM	D_{1M}	0.756	0.670	0.673
							D_{2M}	0.988	0.986	0.987
							D_{JM}	0.914	0.917	0.915
D_{2S}	RESNET50	LLT	6	180 - 146 - 112 - 98 - 58 - 32	0.13 - 0.13 - 0.12 - 0.10 - 0.12 - 0.10	SGD	D_1S	0.744	0.655	0.658
							D_{2S}	0.972	0.972	0.972
							D_{JS}	0.784	0.782	0.783
D_{JI}	VGG19	LLT	6	216 - 137 - 124 - 64 - 48 - 16	0.15 - 0.15 - 0.15 - 0.10 - 0.12 - 0.10	SGD	D_{1I}	0.912	0.914	0.913
							D_{2I}	0.893	0.892	0.892
							D_{JI}	0.947	0.951	0.949
D_{JM}	RESNET50	LLT	5	104 - 104 - 50 - 50 - 20	0.25 - 0.10 - 0.14 - 0.13 - N	NADAM	D_{1M}	0.902	0.899	0.900
							D_{2M}	0.865	0.863	0.864
							D_{JM}	0.967	0.971	0.969
D_{JS}	VGG19	LLL	4	122 - 118 - 64 - 24	0.18 - N - 0.12 - 0.10	RMSProp	D_1S	0.858	0.852	0.855
							D_{2S}	0.880	0.875	0.877
							D_{JS}	0.948	0.951	0.950

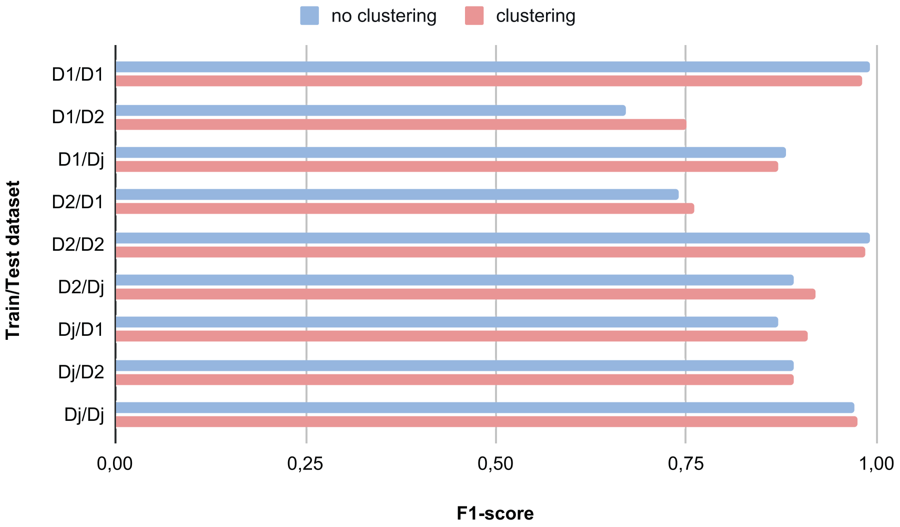

Fig. 9. Comparison between clustering and no clustering in different training and testing scenarios of the best single neural networks.

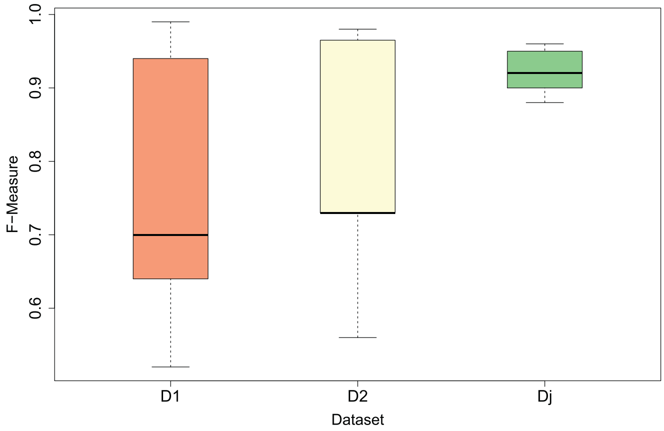

Fig. 10. Average F1-score for the single classifiers trained on D_1 (orange box), D_2 (yellow box), and D_J (green box). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 4
Kolmogorov-Smirnov test comparing the statistical significance, with a confidence of 1%, of the average F1-score distributions in different training and testing conditions.

Tests		p-value	effect size
D_1	D_2	>0.01	-
D_J	D_1	<0.01	0.66 (medium)
D_J	D_2	<0.01	0.62 (medium)

usually contributes to improve the performance of the F1-score. For example, looking at the classifiers trained on D_1 in Table 2, we observe that the best F1-score is 0.994, obtained when the test is performed on D_1. Looking at the classifiers trained and tested on the three clusters of D_1, we obtain the best result (0.975) for the combination (D_{1M}, D_{1M}).

Conversely, looking at the classifiers trained on D_1 in Table 2, we observe that the best F1-score is 0.687 obtained when the test is performed on D_2, but, in the case of clustering, this increases to 0.776 training on D_{1I} and testing on D_{2I}. This result is not surprising since the adoption of the clustering reduces for each classifier the number of cases, while increasing the similarity between images.

Finally, the results of Table 3 confirm again that the best results are obtained training and testing on the same dataset.

With reference to RQ3, Table 2 shows the performance obtained by the single classifiers using the integrated dataset (D_J) for training as well. The data show that the convolutional neural networks trained on D_J leads, in all cases, to a better performance when different train and test sets are used.

In order to better explain this result, the average F1-score is reported and compared in Fig. 10, when D_1, D_2, and D_J are used as training datasets, respectively. The box-plots show that the adoption of the integrated dataset leads to better results: training on D_J has a better average F1-score and the least dispersion effects around it. In Table 4 we show the results of the non-parametric Kolmogorov-Smirnov test over the average F1-score distributions of the three considered cases (i.e., training on D_1, training on D_2, and training on D_J). The results show that there is no statistical difference (p-value greater than a 1% of confidence) between the training on D_1 and training on D_2 experiments, whereas there is a statistical difference (p-value smaller than a 1% of confidence) between the training on D_J and the training on D_1 or between the training on D_J and the training on D_2. This confirms that, when training on the integrated dataset D_J, the resulting classifiers are more robust.

Table 5
Results for the ensemble classifier trained on the integrated dataset as well as on the single datasets.

Training	**Test**	**P**	**R**	**F1**
colrule D_1	D_1	0.975	0.974	0.975
D_1	D_2	0.819	0.822	0.820
D_1	D_J	0.812	0.812	0.812
D_2	D_1	0.797	0.795	0.796
D_2	D_2	0.986	0.986	0.986
D_2	D_J	0.909	0.911	0.910
D_J	D_1	0.975	0.975	0.975
D_J	D_2	0.978	0.945	0.961
D_J	D_J	0.998	0.994	0.996

Table 6
Performance of the best single classifier and of the ensemble trained on D_J and tested on D_3.

Test	Classifier	P	R	**F1**
D_3	ResNet50/D_J	0.856	0.682	0.759
D_3	Ensemble/D_J	0.893	0.913	0.903

Finally, concerning RQ4, Table 5 shows the performance of the ensemble classifier on D_1, D_2, and D_J. The data in the table confirm that the best results are obtained when the network is trained on D_J.

In Fig. 12, the F1-score of the best single classifiers (as reported in Table 2) and the ensemble classifier, on different combinations of the train and test datasets, are compared.

It is possible to observe that the ensemble generally outperforms the best single classifier and this takes place always whenever the training dataset is the integrated one. Moreover, in this case, the performance of the ensemble trained on the integrated dataset is very stable, given that the F1-score ranges from 0.94 to 0.95.

This is also confirmed by looking at accuracy and loss trends over the training epochs: the ensemble classifier performs better and reaches a higher final accuracy, also requiring fewer training epochs to obtain the same performance. This is clearly shown in Fig. 11[9], representing the comparison of Accuracy (left) and Loss (right) versus the epochs of the ensemble and VGG19 classifiers. More precisely, the figure shows i) the ensemble classifier performance when D_J is used for both training and testing (blue curve), ii) the ensemble classifier performance when D_J is used for training and D_2 for testing (red curve), iii) the VGG19 performance when D_J is used for both training and testing (black curve). From the figure, it is evident that the ensemble classifier performs better than VGG19 and even better whenever the integrated dataset is used for both training and testing.

In order to perform a more strict validation of the proposed approach, an additional dataset D_3 was used. D_3 was not used for training, but just for validating the ensemble classifier results. Therefore, Table 6 and Fig. 13 compare the performance of the best single classifier (RESNET50 in this case) and the ensemble on this additional dataset used as test set. The training dataset is in both cases D_J. It clearly emerges that when the additional dataset is tested, the classification performance decreases. However, this is due to the fact that the new dataset has different characteristics (i.e., resolution, format, gray-scale profile) with respect to the ones used for the training. Nevertheless, also in this situation the ensemble classifier largely outperforms the single best classifier, with an F1-score equal to 0.903. This confirms that the proposed approach provides better results when used in real contexts.

[9] The first dataset in the legend refers to the training, while the second one to the test.

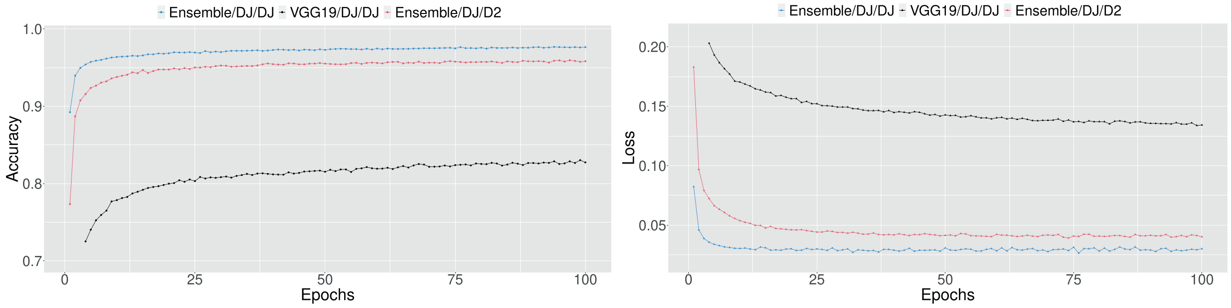

Fig. 11. Performance comparison among the ensemble and the best pre-trained classifiers during training: validation accuracy and loss over epochs.

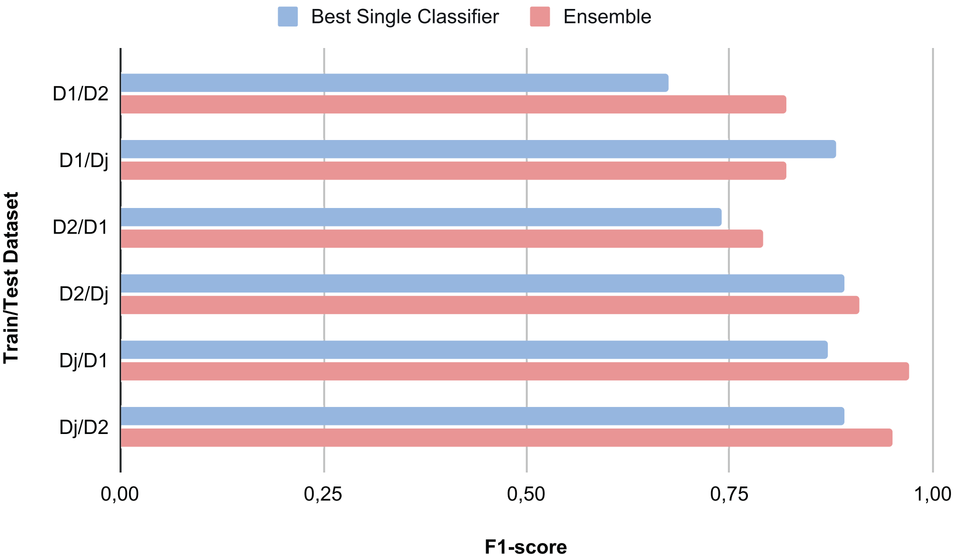

Fig. 12. F1-score of the best single classifiers compared with the ensemble classifiers on the different datasets.

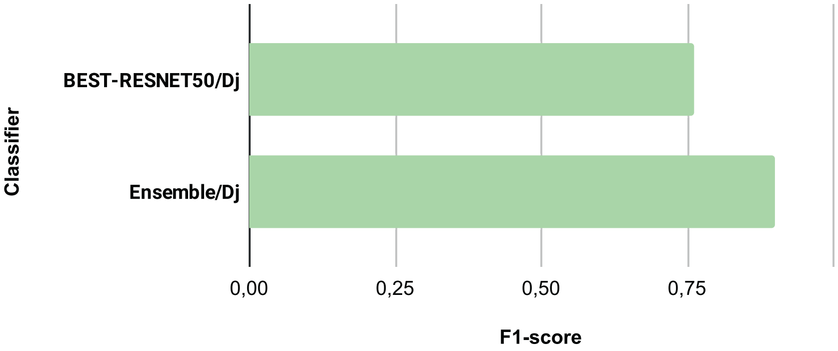

Fig. 13. F1-score of the best single classifier compared with the ensemble classifier, while testing on dataset D_3 and training on dataset D_J.

Finally, during the preliminary set-up of the experiments, we also analyzed the contributions of the individual networks to the results of the ensemble approach. It emerged that the best single classifier trained on the Medium Lobe cluster (i.e., VGG19) obtained outcomes closer to the final results of the ensemble than the other networks.

For what concerns training times, they basically depend on three factors: (i) the network model used as a basis for the solution; (ii) the dataset on which the model is re-trained; (iii) the considered re-training specification, as models with a higher number of re-trained blocks require more training effort. On the GPUs adopted in the proposed experimentation, the average training time of a single configuration for 15 epochs ranges from 815s ($\simeq$15.3 min), for ResNet50 on D_2 with all layers untouched, to 3585s ($\simeq$57 min) for VGG19 on D_J with all layers re-trained.

To have a more precise picture of the impact of the pre-trained models over the final performance, we studied the performance of the single classifiers using different types of CNNs (i.e., VGG 16/19, Resnet 50/101/151, Xception, Inception V2/V3). Specifically, Fig. 14 shows the performance, considering the F1-score of the best configurations produced by the evolutionary algorithm execution, in decreasing order. This test was made to find out the best sub-types of CNNs, which resulted to be ResNet50, VGG19, and Xception as per the figure. Only in the case of ResNets, the performance was equivalent for both ResNet50 and ResNet101. Being almost equivalent in terms of performance, we selected ResNet50 for mainly two reasons: (*i*) it converges to a better configuration more often

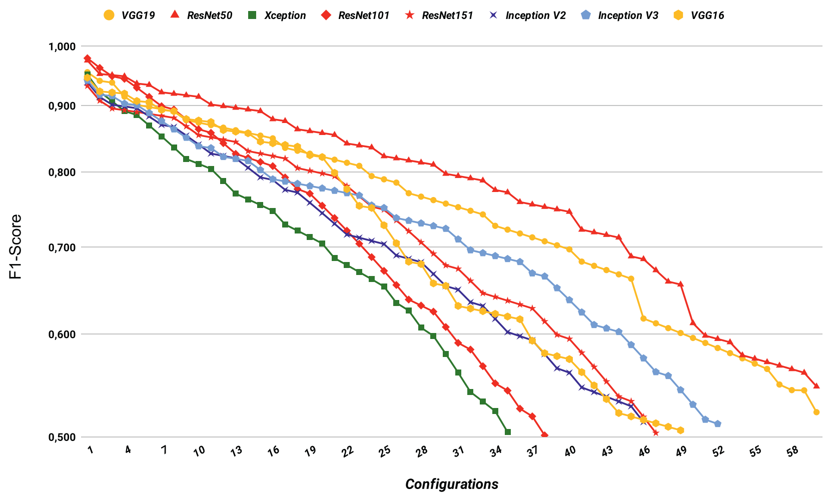

Fig. 14. F1-score performance over various configurations of the genetic algorithm for different types of CNNs for the single classifier.

than ResNet101, thus indicating a greater stability for the specific task and (*ii*) it is better in terms of training time having a simpler structure with a lower number of layers.

7. Threats to the validity

This section discusses the threats to the validity of the described study. We identified different types of threats: construct, internal, and external validity.

As regards the construct validity threats, some limitations can be due to the reduced number of datasets used to build the joint dataset: only two datasets were used for the integration process. In order to avoid this limitation, additional datasets should be integrated. However, at the moment of performing this study, other public datasets, which can be integrated with the two we considered, are not available. In Section 5 we listed the requirements of the selected datasets.

Internal validity threats concern variables internal to our study, and not considered in our experiment, that could influence our observations on the dependent variable. In our study, we always split data into 80% training set and 20% test set. However, it is possible that different splits could produce different results. Moreover, as our results already confirmed, the outcomes can vary based on the choice of a specific network used for the prediction. We cannot exclude that other models not considered in our study could exhibit different and perhaps better performance. In addition, in the proposed study the adopted datasets were automatically clustered and labeled; therefore, we could have clustering errors. However, to mitigate this risk, a manual clustering for the whole integrated dataset was performed, in order to assess the automated clustering results and obtain a more rigorous process. Moreover, it is necessary to underline that the used datasets are well documented and referenced in medical studies and a domain expert was involved during the manual clustering assessment.

Finally, threats to external validity refer to the possibility to generalize the obtained outcomes. We have evaluated our approach on a relevant number of CT scan images coming from two existing real-world datasets. Moreover, we have also evaluated our approach on an additional dataset having different image formats, resolution, and colors; therefore, this type of threat should be limited.

8. Conclusions

In this paper, we have proposed an innovative approach that aims at detecting patients affected by Covid-19 using CT scan images. The approach is defined considering computerized tomography images clustered into three main sections of a lung (superior, middle, and inferior lobe) used to train both single state-of-the-art convolutional neural networks and a hierarchical multiple classifier composed of them. Moreover, we have applied a genetic algorithm to select the best performing pre-trained models, by optimizing both their hyper-parameters and the internal structure of the convolutional neural networks with an automatic approach (AutoML). In order to validate such an approach, a large dataset has been constructed through the integration of two open existing datasets. After optimizing the single classifiers, implemented using a transfer learning approach, we have built a hierarchical multiple classifier using the majority strategy for the final voting procedure. We have compared the results obtained by the single classifiers and the ensemble classifier demonstrating that the ensemble overall outperforms the best single classifiers, provided that it is trained on the integrated dataset. Moreover, the performance of the ensemble trained on the integrated dataset is very stable, given that the F1-score ranges only from 0.94 to 0.95.

As highlighted in the previous section, the main limitation of this study is the reduced number of datasets used to build the joint dataset. According to this, as future work, new datasets will be considered and integrated to extend our investigation and to obtain an even larger dataset. Moreover, we plan to test other allocation strategies and other voting techniques for the ensemble classifier itself.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, T. Chen, Recent advances in convolutional neural networks, Pattern Recognit. 77 (2018) 354–377, doi:10.1016/j.patcog.2017.10.013.

[2] K.C. Santosh, AI-driven tools for coronavirus outbreak: need of active learning and cross-population train/test models on multitudinal/multimodal data, J. Med. Syst. 44 (5) (2020) 93, doi:10.1007/s10916-020-01562-1.

[3] M. Chung, A. Bernheim, X. Mei, N. Zhang, M. Huang, X. Zeng, J. Cui, W. Xu, Y. Yang, Z.A. Fayad, et al., CT imaging features of 2019 novel coronavirus (2019-nCoV), Radiology 295 (1) (2020) 202–207.

[4] X. Wang, X. Jiang, H. Ding, Y. Zhao, J. Liu, Knowledge-aware deep framework for collaborative skin lesion segmentation and melanoma recognition, Pattern Recognit. (2021) 108075, doi:10.1016/j.patcog.2021.108075.

[5] L. Aversano, M.L. Bernardi, M. Cimitile, R. Pecori, Early detection of parkinson disease using deep neural networks on gait dynamics, in: 2020 International

Joint Conference on Neural Networks, IJCNN 2020, Glasgow, United Kingdom, July 19–24, 2020, IEEE, 2020, pp. 1–8, doi:10.1109/IJCNN48605.2020.9207380.

[6] I.D. Apostolopoulos, T.A. Mpesiana, COVID-19: automatic detection from x-ray images utilizing transfer learning with convolutional neural networks, Phys. Eng. Sci. Med. 43 (2) (2020) 635–640, doi:10.1007/s13246-020-00865-4.

[7] M. Shervin, K. Rahele, S. Milan, Y. Shakib, S. Ghazaleh Jamalipour, Deep-covid: predicting COVID-19 from chest x-ray images using deep transfer learning, Med. Image Anal. 65 (2020) 101794, doi:10.1016/j.media.2020.101794.

[8] Sakshi Ahuja S., Deep transfer learning-based automated detection of COVID-19 from lung CT scan slices, Appl. Intell. (2020), doi:10.1007/s10489-020-01826-w.

[9] S. Hu, Y. Gao, Z. Niu, Y. Jiang, L. Li, X. Xiao, M. Wang, E.F. Fang, W. Menpes–Smith, J. Xia, H. Ye, G. Yang, Weakly supervised deep learning for COVID-19 infection detection and classification from ct images, IEEE Access 8 (2020) 118869–118883.

[10] S.A. Harmon, T.H. Sanford, S. Xu, E.B. Turkbey, H. Roth, Z. Xu, D. Yang, A. Myronenko, V. Anderson, A. Amalou, M. Blain, M. Kassin, D. Long, N. Varble, S.M. Walker, U. Bagci, A.M. Ierardi, E. Stellato, G.G. Plensich, G. Franceschelli, C. Girlando, G. Irmici, D. Labella, D. Hammoud, A. Malayeri, E. Jones, R.M. Summers, P.L. Choyke, D. Xu, M. Flores, K. Tamura, H. Obinata, H. Mori, F. Patella, M. Cariati, G. Carrafiello, P. An, B.J. Wood, B. Turkbey, Artificial intelligence for the detection of COVID-19 pneumonia on chest CT using multinational datasets, Nat. Commun. 11 (1) (2020) 4080, doi:10.1038/s41467-020-17971-2.

[11] P. Silva, E. Luz, G. Silva, G. Moreira, R. Silva, D. Lucio, D. Menotti, COVID-19 detection in CT images with deep learning: avoting-based scheme and cross-datasets analysis, Inf. Med. Unlocked 20 (2020) 100427, doi:10.1016/j.imu.2020.100427.

[12] J. Ahmad, H. Farman, Z. Jan, Deep Learning Methods and Applications, Springer Singapore, Singapore, 2019, pp. 31–42.

[13] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: Y. Bengio, Y. LeCun (Eds.), 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings, 2015.

[14] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.

[15] F. Chollet, Xception: deep learning with depthwise separable convolutions, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 1800–1807.

[16] I. Loshchilov, F. Hutter, CMA-ES for hyperparameter optimization of deep neural networks, 2016, 1604.07269

[17] M.A. Albadr, S. Tiun, M. Ayob, F. AL-Dhief, Genetic algorithm based on natural selection theory for optimization problems, Symmetry 12 (11) (2020), doi:10.3390/sym12111758.

[18] L. Aversano, M.L. Bernardi, M. Cimitile, R. Pecori, Fuzzy neural networks to detect parkinson disease, in: 29th IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2020, Glasgow, UK, July 19–24, 2020, IEEE, 2020, pp. 1–8, doi:10.1109/FUZZ48607.2020.9177948.

[19] Z. Hu, J. Tang, Z. Wang, K. Zhang, L. Zhang, Q. Sun, Deep learning for image-based cancer detection and diagnosisa survey, Pattern Recognit. 83 (2018) 134–149, doi:10.1016/j.patcog.2018.05.014.

[20] A.S. Lundervold, A. Lundervold, An overview of deep learning in medical imaging focusing on MRI, Zeitschrift fr Medizinische Physik 29 (2) (2019) 102–127, doi:10.1016/j.zemedi.2018.11.002. Special Issue: Deep Learning in Medical Physics

[21] S. Roy, W. Menapace, S. Oei, B. Luijten, E. Fini, C. Saltori, I. Huijben, N. Chennakeshava, F. Mento, A. Sentelli, E. Peschiera, R. Trevisan, G. Maschietto, E. Torri, R. Inchingolo, A. Smargiassi, G. Soldati, P. Rota, A. Passerini, R.J.G. van Sloun, E. Ricci, L. Demi, Deep learning for classification and localization of COVID-19 markers in point-of-care lung ultrasound, IEEE Trans. Med. Imaging 39 (8) (2020) 2676–2687.

[22] H. Mukherjee, S. Ghosh, A. Dhar, S.M. Obaidullah, K.C. Santosh, K. Roy, Deep neural network to detect COVID-19: one architecture for both CT scans and chest x-rays, Appl. Intell. (2020), doi:10.1007/s10489-020-01943-6.

[23] V. Positano, A.K. Mishra, S.K. Das, P. Roy, S. Bandyopadhyay, Identifying COVID-19 from chest CT images: a deep convolutional neural networks based approach, J. Healthc Eng. 2020 (2020) 8843664.

[24] 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, J. Xia, Artificial intelligence distinguishes COVID-19 from community acquired pneumonia on chest CT, Radiology (2020).

[25] A.M. Hasan, M.M. AL-Jawad, H.A. Jalab, H. Shaiba, R.W. Ibrahim, A.R. AL-Shamasneh, Classification of COVID-19 coronavirus, pneumonia and healthy lungs in CT scans using q-deformed entropy and deep learning features, Entropy 22 (5) (2020), doi:10.3390/e22050517.

[26] A.A. Ardakani, A.R. Kanafi, U.R. Acharya, N. Khadem, A. Mohammadi, Application of deep learning technique to manage COVID-19 in routine clinical practice using CT images: results of 10 convolutional neural networks, Comput. Biol. Med. 121 (2020) 103795, doi:10.1016/j.compbiomed.2020.103795.

[27] X. Mei, H.-C. Lee, K.-y. Diao, M. Huang, B. Lin, C. Liu, Z. Xie, Y. Ma, P.M. Robson, M. Chung, A. Bernheim, V. Mani, C. Calcagno, K. Li, S. Li, H. Shan, J. Lv, T. Zhao, J. Xia, Q. Long, S. Steinberger, A. Jacobi, T. Deyer, M. Luksza, F. Liu, B.P. Little, Z.A. Fayad, Y. Yang, Artificial intelligence–enabled rapid diagnosis of patients with COVID-19, Nat. Med. 26 (8) (2020) 1224–1228, doi:10.1038/s41591-020-0931-3.

[28] D. Colombi, F.C. Bodini, M. Petrini, G. Maffi, N. Morelli, G. Milanese, M. Silva, N. Sverzellati, E. Michieletti, Well-aerated lung on admitting chest CT to predict adverse outcome in COVID-19 pneumonia, Radiology 296 (2) (2020) E86–E96, doi:10.1148/radiol.2020201433.

[29] T. Zhou, H. Lu, Z. Yang, S. Qiu, B. Huo, Y. Dong, The ensemble deep learning model for novel COVID-19 on CT images, Appl. Soft Comput. 98 (2021) 106885, doi:10.1016/j.asoc.2020.106885.

[30] S. Shastri, K. Singh, S. Kumar, P. Kour, V. Mansotra, Deep-LSTM ensemble framework to forecast COVID-19: an insight to the global pandemic, Int. J. Inf. Technol. (2021), doi:10.1007/s41870-020-00571-0.

[31] P. Gifani, A. Shalbaf, M. Vafaeezadeh, Automated detection of COVID-19 using ensemble of transfer learning with deep convolutional neural network based on CT scans, Int. J. Comput. Assist. Radiol. Surg. 16 (1) (2021) 115–123, doi:10.1007/s11548-020-02286-w.

[32] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, L. Fei-Fei, ImageNet large scale visual recognition challenge, Int. J. Comput. Vis. 115 (3) (2015) 211–252, doi:10.1007/s11263-015-0816-y.

[33] C. Li, A.C. Bovik, Content-partitioned structural similarity index for image quality assessment, Signal Process. Image Commun. 25 (7) (2010) 517–526, doi:10.1016/j.image.2010.03.004. Special Issue on Image and Video Quality Assessment

[34] G. Renieblas, A. Turrero, A. Muoz, N. Len, E. Guibelalde, Structural similarity index family for image quality assessment in radiological images, J. Med. Imaging 4 (2017) 035501, doi:10.1117/1.JMI.4.3.035501.

[35] S. Lloyd, Least squares quantization in PCM, IEEE Trans. Inf. Theory 28 (2) (1982) 129–137, doi:10.1109/TIT.1982.1056489.

[36] H. Zhou, J. Gao, Automatic method for determining cluster number based on silhouette coefficient, Adv. Mat. Res. 951 (2014) 227–230.

[37] R. Llet, M. Ortiz, L. Sarabia, M. Snchez, Selecting variables for k-means cluster analysis by using a genetic algorithm that optimises the silhouettes, Anal. Chim. Acta 515 (1) (2004) 87–100, doi:10.1016/j.aca.2003.12.020.

[38] H.G. Ayad, M.S. Kamel, On voting-based consensus of cluster ensembles, Pattern Recognit. 43 (5) (2010) 1943–1953, doi:10.1016/j.patcog.2009.11.012.

[39] Y. Kokkinos, K.G. Margaritis, Breaking ties of plurality voting in ensembles of distributed neural network classifiers using soft max accumulations, in: L. Iliadis, I. Maglogiannis, H. Papadopoulos (Eds.), Artificial Intelligence Applications and Innovations, Springer Berlin Heidelberg, Berlin, Heidelberg, 2014, pp. 20–28.

[40] I. Gitman, H. Lang, P. Zhang, L. Xiao, Understanding the role of momentum in stochastic gradient methods, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems, vol. 32, Curran Associates, Inc., 2019.

Lerina Aversano is an associate professor at the Department of Engineering of the University of Sannio Benevento (Italy). She received the Ph.D. in Computer Engineering in July 2003 at the same University where she has been assistant professor from 2005. She also was a research leader at RCOST - Research Centre On Software Technology - of the University of Sannio from 2005. She published more than 80 papers in journals and conference proceedings. Her research interests include software maintenance, program comprehension, reverse engineering, reengineering, migration, business process modelling, business process evolution, software system evolution, software quality.

Mario Luca Bernardi received the Laurea degree in Computer Science Engineering from the University of Naples "Federico II", Italy, in 2003 and the Ph.D. degree in Information Engineering from the University of Sannio in 2007. He is currently an assistant professor of Computer Science at the University of Sannio. Since 2003 he has been a researcher in the field of software engineering and his list of publications contains more than 60 papers published in journals and conference proceedings. His main research interests include software engineering (maintenance, testing, business process management, reverse engineering and data mining on software systems, software quality assurance with particular interest on internal quality metrics and on new paradigms for software modularity, including aspect-oriented software, component-based software and model-driven development). He serves both as a member of the program and organizing committees of conferences, and as associate editor and reviewer of papers submitted to some of the main journals and magazines in the field of software engineering, software maintenance and program comprehension.

Marta Cimitile is Assistant Professor and Aggregated Professor at Unitelma Sapienza University of Rome (Italy). She received a PhD in Computer Science in 06/05/2008 at the Department of Computer Science at the University of Bari and she received the Italian Scientific Qualification for the Associate Professor position in Computer Science Engineering in April 2017. She published more than fifty papers at international conferences and journals. Her main research topics are: Business Process Management and modeling, Knowledge modeling and Discovering, Process and Data Mining in Software Engineering Environment. In the last year, she was involved in several industrial and research projects and she is a founding member of the SpinOff of University of Bari named Software Engineering Research and Practices s.r.l. (www.serandp.com). She was in the program and organizing committees of several international conferences, she is reviewer to some of the main journals and magazines in the field of Knowledge Management and Software Engineering, knowledge representation and transfer and data mining and she is in the Editorial Board of the Journal of Information and Knowledge Management, PeerJ Computer Science. She is IEEE Member and IEEE Access reviewer.

Riccardo Pecori got his Ph.D. in Information Technology from University of Parma in 2011. From May 2015 to July 2019 he has been Assistant Professor of Computer Science at eCampus University where he taught Computer Security, Network Security, Internet of Things and Information Technology for Psychologists. Since August 2019 he is Assistant Professor at University of Sannio teaching ”Digital Design”. Since April 2017 he has been editor of the journal ”Future Generation Computer Systems” and since September 2019 of journal ”SoftwareX”. He has been Program Chair of WIVACE 2018 and HELMeTO 2019, organizing also a special session on “Social Internet of Things” at ISWCS 2017. His research interests regard network security, educational and social Big Data analysis, and identification of relevant sets in complex systems as well as deep learning.