Journal of Universal Computer Science, vol. 30, no. 6 (2024), 758-778
submitted: 19/9/2023, accepted: 13/1/2024, appeared: 28/6/2024

Deep learning techniques to process 3D chest CT

Mauricio Solar
(Universidad Tecnica Federico Santa Maria, Santiago, Chile
https://orcid.org/0000-0002-4433-4622, mauricio.solar@usm.cl)

Pablo Aguirre
(Universidad Tecnica Federico Santa Maria, Santiago, Chile
pablo.aguirre.14@sansano.usm.cl)

Abstract: The idea of using X–rays and Computed Tomography (CT) images as diagnostic method has been explored in several studies. Most of these studies work with slices of CT image in 2D, requiring less computational capacity and less time to process them than 3D. The processing of volumetric data (the complete CT images in 3D) adds an extra dimension of information. However, the magnitude of the data is considerably larger than working with slices in 2D, so extra computational processing is required. In this study a model capable of performing a classification of a 3D input that represents the volume of the CT scan is proposed. The model is able to classify the 3D input between COVID–19 and Non–COVID–19, but reducing the use of resources when performing the classification. The proposed model is the *ResNet–50* model with a new dimension of information added, which is a simple *autoencoder*. This *autoencoder* is trained on the same dataset, and a vector representation of each exam is generated and used together with the exams to feed the *ResNet–50*. To validate the proposal, the same proposed model is compared with and without the *autoencoder* module that provides more information to the proposed model. The proposed model obtains better metrics than the same model without the *autoencoder*, confirming that extracting relevant features from the dataset helps improve the performance of the model.

Keywords: CNN, COVID–19, deep learning, CT
Categories: H.3.1, H.3.2, H.3.3, H.3.7, H.5.1
DOI: 10.3897/jucs.112977

1 Introduction

COVID–19, with more than 80 million cases confirmed worldwide and more than 1.8 million deaths [WHO, 2020], caused the collapse of health systems worldwide prior to the development of a vaccine. To alleviate this situation, the use of Artificial Intelligence (AI) has been proposed to accelerate the process of diagnosis and consequent treatment of this disease symptoms-related patients [Kaya et al., 2022c].

The idea of using Computed Tomography (CT) imaging as an alternative diagnostic method has been explored in several studies. CT is a test that combines a series of X–rays taken from different angles around the body which are then processed to create cross-sectional images of the bones and soft tissues, among others. In this way, it is possible to work with slices of CT image in 2D, requiring less computational capacity and time to process them than 3D ([Eken, 2023], [Kaya et al., 2022a], [Kaya et al., 2022b], [Kaya et al., 2022c], [Yilmaz, 2021], [Sethy et al., 2020], [Dansana et al., 2023], [Gaur et at., 2023])

The processing of volumetric data, i.e. the complete CT images in 3D, adds an extra dimension of information. However, the magnitude of the data is considerably larger than working only with slices in 2D, so extra computational processing is required [Nair et al., 2021], [Wang et al., 2020], [Li et al., 2020].

CT images are mainly found in DICOM format, an international standard for transmitting, storing, retrieving, processing and presenting medical imaging information. It consists of a series of objects with their respective attributes [Pianykh, 2009]. These include the parameters of the equipment used to capture the medical images, the actual images and the patient information. Therefore, a pre-processing is required to separate the images from the diagnoses.

The motivation of this work is to propose a model that is capable of performing a classification with a 3D input that represents the volume of the CT scan. Since working on 2D X-ray images consumes a large amount of resources, processing 3D volumes further stresses resource consumption, making a practical solution unfeasible. This motivate us to propose a model that is capable of performing a classification with a 3D input, but reducing the use of resources when performing the classification.

Most of the works found in the literature apply slice-level (2D) diagnostics and use techniques such as segmentation masks and extraction of regions of interest to enrich the dataset, which is resource-intensive. This proposal is based on the assumption that deep learning models have better performance if they have access to a more complete dataset and not because of the processing (which consumes resources) that is carried out on the dataset.

In this sense, the classification of the proposed model is carried out at the exam level, with the input of the model being 3D volumes. The modifications made to the dataset are: (1) reduce dimensions to reduce the use of resources, and (2) rotate the exams so that they have the same perspective since, unlike working with slices, a top view is very different from a side view which could introduce noise.

In search of the solution, a first *ResNet-50* model was implemented, whose performance was lower than those presented in the literature, either because the size of the images is smaller than that used in the other studies or because the previous processing of the data they made is relevant.

Since the datasets used are different and the processes applied in some works are highly complex, it is not easy to compare results with those in the literature. To validate if the proposal has good behavior, the same proposed model is compared with and without the *autoencoder* module that provides more information to the proposed model. That is, the proposed model is the *ResNet–50* model to which a new dimension of information is added, which is a simple *autoencoder* [Badr, 2019]. With this *autoencoder* trained on the same dataset, a vector representation of each exam is generated and used together with the exams to feed the *ResNet–50*. The greatest number of variables is kept fixed to isolate the contribution of this additional information while maintaining low complexity.

The proposed model obtains better metrics than the same model without the *autoencoder*, confirming that extracting relevant features from the dataset helps improve the performance of the model.

The solution proposed in this article consists of a *deep learning* model that can identify CT images affected by COVID–19 without the need of a sophisticated task to make it work. For this purpose, a modular architecture is developed to work with 3D data, taking advantage of the correlated information between cuts when deciding.

The contributions of this work are as follows: (i) Introduced an approach based on *ResNet–50* model with an *autoencoder* module that can classify the COVID–19 and Non–COVID–19 using 3D CT scans images for the prediction of COVID–19. (ii) Pro-

posed model has been evaluated on the basis of metrics such as accuracy, precision, recall, specificity and AUC. (iii) Finally, the proposed model with the *autoencoder* module has been compared with the same proposed model without the *autoencoder* module.

The following section presents the conceptual framework, and then we find some proposed solutions successfully raised by the state of the art, solving the problem of diagnosing COVID–19 in CT chest images. The section 'Proposed Model' presents the proposed solution, the motivation, data used and details of the *deep learning* architecture. Then, the section presents an analysis and a comparison with those proposals in the state of the art. Finally, the conclusions section summarizes the work carried out.

2 Conceptual Framework

2.1 Computed Tomography (CT) Images

CT operates in a similar way to X–rays. Different parts of the body absorb different amounts of radiation, so CT images show an appreciable contrast.

The procedure consists of a series of X–ray beams and electronic detectors that rotate around the patient. These detectors measure the amount of radiation that is absorbed by the entire body. Then, a software processes the volume of data generated to create 2D cross-sectional images of the body.

Advances in technology have enabled CT scanners to obtain multiple slices of the body in a single rotation. Furthermore, slices can be finer, resulting in increased detail and additional viewing capabilities.

2.2 Use of CT in COVID–19 Diagnosis

According to the *Fleischner Society*, CT imaging is not indicated as a diagnostic test of COVID–19 in asymptomatic patients or those with mild respiratory symptoms [Rubin et al., 2020]. Several studies have been published reporting findings of COVID–19 on CTs of the thorax [Adams et al., 2020]. Lung histologic findings of COVID-19 are characterized by alveolar diffuse damage and resemble those observed in other coronaviruses, such as SARS–CoV–1 and MERS–CoV [Schaller et al., 2020]. The prevalence of COVID–19–related abnormalities on CT images depends on the stage it is in, and its severity [Kwee and Kwee, 2020, Adams et al., 2020, Shah et al., 2020].

In [Kaya et al., 2022b] is proposed a method to apply the angle transform (AT) method to the X-ray images. This transformation uses the angle information created by each pixel on the image with the surrounding pixels obtaining eight different images for each image in the dataset. These images are trained with a hybrid deep learning model, which combines GoogleNet and long short-term memory (LSTM) models, and COVID-19 disease detection is carried out with a high classification accuracy of 98.97% with this approach. The AT + GoogleNet + LSTM approach is successful for COVID-19 detection using chest X-ray images.

[Eken, 2023] shows a hierarchical middleware for COVID-19 detection in X-ray image and its metadata.

In [Yilmaz, 2021] the COVID-19 is diagnosed automatically from X-ray images using a multi-channel CNN method. The proposal has with five convolution channels and the channel selection formulas is used for selecting the most distinctive feature filters among the results produced by these channels.

Several other studies with 2D X-ray images are presented in the literature, such as use of Deep Features and Support Vector Machine in [Sethy et al., 2020], use of VGG16, InceptionV2 and Decision Tree in [Dansana et al., 2023], use of VGG16, InceptionV3 and EfficientNetB0 in [Gaur et at., 2023], etc.

CORNet is a 3D deep learning model based on *ResNet-50* proposed by [Nair et al., 2021]. CORNet performed the retrospective and multicenter analysis for the extraction of visual characteristics from volumetric chest CT scans during COVID-19 detection. [Wang et al., 2020] proposed a 3D deep CNN to detect COVID-19 from CT volumes, named DeCoVNet, which takes as input a CT volume and its 3D lung mask. And this 3D lung mask is generated by a pre-trained UNet.

2.3 *Deep Learning* as a Diagnostic Tool

The rapid growth of high-resolution medical imaging requires more effort on the part of health care professionals when analyzing and perform diagnoses, which turn out to be subjective, error-prone and can vary from one professional to another.

As an alternative, the use of *machine learning* techniques has been proposed to automate the diagnostic process. However, there is a large variability between patients, and these techniques are largely dependent on a large extent of characteristics defined by experts, so they are not sufficient to deal with complex problems. The next step is to let the computers learn to identify the features that best formally represent the data for the problem to be solved. This is the key feature of *deep learning*: models composed of many layers that can be used, for example, to perform a transformation to an *input* (in this case images), obtain an *output* that indicates presence or absence of disease, and in the process, learn relevant features of the data that help in solving the problem.

2.3.1 Artificial Neural Networks

Neural networks, in their simplest form, consist of thousands or millions of highly interconnected processing nodes. These nodes or neurons are organized in unidirectional layers, through which data are spread during training, and each of them can be connected to several nodes located in a previous layer, from where it receives data and several nodes in a subsequent layer transmitting the data after processing.

Each of these node connections, has an associated *weight*. During training, nodes receive data as numbers, they are multiplied by these weights, add all the resulting values to obtain a single figure and if it exceeds a threshold, it is transmitted to the subsequent nodes. For a given l layer, this process is summarized in the equation (1).

$$\begin{aligned} z^{(l)} &= W^{(l)} \cdot a^{(l-1)} + b^{(l)} \\ a^{(l)} &= \sigma(z^{(l)}) \end{aligned} \tag{1}$$

Where $W^{(l)}$ corresponds to the matrix of weights associated to the nodes of layer l, $a^{(l-1)}$ corresponds to the activations of the nodes of the previous layer (or the input data in case it is the first layer), $a^{(l)}$ corresponds to the activations of the current layer, $b^{(l)}$ corresponds to the bias of the nodes and σ corresponds to the activation function that acts as the threshold that determines if the values are propagated to the subsequent layers.

In a neural network, changing the weights of any connection has a cascading effect on the rest of the nodes and their activations in subsequent layers. Therefore, the learning

process that characterizes neural networks occurs, given a set of *inputs* and *outputs*, by obtaining the weights that minimize a cost function according to an optimization algorithm, where the cost function corresponds to an approximation of how wrong the predictions are with respect to the expected values.

This optimization process of the weights is known as *backpropagation*, which allows to calculate the error attributable to each node and therefore, the partial derivatives that form the gradient used in the optimization algorithms as a downward gradient. Each element of the gradient indicates how much the cost function changes if a change is made to that parameter.

2.3.2 Convolutional Neural Networks (CNN)

Convolutional neural networks (CNN) are *deep learning* algorithms composed of convolutional layers that take an image as an *input*, learn the relevance of different aspects or objects within the image and are able to differentiate among them.

Images are matrices where each value is associated with a pixel. Therefore, using neural networks with dense layers is not efficient, since the number of connections between nodes increases considerably, making the learning process slow and costly. A convolutional network is able to efficiently capture spatial and temporal relationships within images due to the reduction in the number of parameters involved during the learning and the reuse of node weights. The first convolutional layers are responsible for extracting global features from the images, while the deep layers are capable of obtaining fine details.

Convolutional layers extract high-level image characteristics by applying convolutions of elements called filters or *kernels*. Filters are matrices that perform convolutions on image segments, by displaying a certain number of pixels at each step, determined by a *stride* parameter until it runs through the entire image.

There are two types of results after performing convolution: in one, the resulting data has a smaller dimension compared to the *input*, while in the other, the dimension is equal or larger. This can be controlled by the application of *padding* on the data before convolutions are made. *Padding* consists of filling with values, usually zeros, the contours of the images to increase their dimensions. There are two types:

- ***Valid Padding***: No *padding* is done on the image, obtaining *outputs* with smaller dimensions.
- ***Same Padding***: *Padding* is done in such a way that the result of the convolution has the same dimensions as the *input*.

There are different uses of CNN methods for diagnostic purposes. An automatic bearing fault size diagnosis using time-frequency images of CWT and deep transfer learning methods is presented in [Kaya et al., 2022a]. Kaya2022b shows an approach for congestive heart failure and arrhythmia classification using angle transformation with LSTM. In [Akdag et al., 2022] an approach for congestive heart failure and arrhythmia classification using downsampling local binary patterns with LSTM is proposed.

2.3.3 Metrics

Metrics to evaluate the performance of prediction models allow to understand how close they are to the desired behavior and provide a framework of comparison with other

models. For classification problems, it is common to use the metrics detailed below. These metrics are derived from the confusion matrix, which is a table that visualizes the performance of a predictive model (Figure 1).

		Predicted	
		Negative	**Positive**
Actual	**Negative**	True Negative	False Positive
	Positive	False Negative	True Positive

Figure 1: Confusion Matrix.

The ***accuracy*** represents the incidence between correct predictions (True Negative-TN + True Positive-TP) and total predictions (True and False both Positive and Negative). It may not be a good metric if the data set is not balanced, i.e., if the number of examples of each class is too different. Its values are in the range between 0 and 1, and can also be expressed as a percentage. It is calculated according to the equation (2).

$$Accuracy = \frac{TN + TP}{TN + FP + TP + FN} \tag{2}$$

Precision represents the incidence of positive predictions (TP) that were correctly classified. Its values are in the ranges from 0 to 1, where values close to 1 indicate that the number of FP is very small compared to TP. It is calculated according to the equation (3).

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

Recall measures the incidence of positive cases that were correctly classified. Its values are in the range between 0 and 1. Values close to 1 indicate that there are few positive cases that were misclassified. It is calculated according to equation (4).

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

Specificity measures the incidence of negative cases that were correctly classified. Its values are in the range between 0 and 1. Values close to 1 indicate that there are few negative cases that were misclassified. It is calculated according to the equation (5).

$$Specificity = \frac{TN}{TN + FP} \tag{5}$$

The ***Negative Predictive Value*** **(NPV)** represents the incidence of negative predictions that were correctly classified. Its values are in the range between 0 and 1, where values close to 1 indicate that the number of FN is very small compared to TN. It is calculated according to the equation (6).

$$NPV = \frac{TN}{TN + FN} \tag{6}$$

The **AUC** corresponds to the value of the area under the **ROC** curve. The ROC curve is a graph showing the performance of a classification model through the comparison between the true positive rate (TPR) and the false positive rate (FPR) overall classification thresholds. The TPR is known as ***Recall***, while the FPR is calculated according to the equation (7).

$$FPR = \frac{FP}{FP + TN} \tag{7}$$

Therefore, the **AUC** can be interpreted as the ability of the predictive model to correctly differentiate between a positive and a negative case. Its values are between 0 and 1, where those close to 1 indicate a higher differentiation capability. The ***F1-score*** corresponds to the harmonic media between *accuracy* and *recall*. Its values are in the range between 0 and 1, where values close to 1 indicate that the *accuracy* and *recall* are very good. It is calculated according to the equation (8).

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{8}$$

3 The State of the Art

Deep Learning techniques are usually used today in several kind of diagnostic models, such as congestive heart failure and arrhythmia classification in Kaya2022a, Kaya2022b and Akdag2022, or lung cancer classification Cengil2018. There are two ways to approach the COVID-19 diagnostic problem using CTs:

- The first consists of working with slices of CT image (2D) because less computational capacity and time is required to process them. Noteworthy are the proposals of [Wang et al., 2021] and [Song et al., 2021].
- On the other hand, this proposal is to work with the complete CT images (3D), i.e., with volumetric data. One advantage of this is that an extra dimension of information is added. However, the magnitude of the data is considerably larger than working with 2D slices, so extra processing is required. The proposals of [Subramaniyan et al., 2021, Nair et al., 2021, Wang et al., 2020] are noteworthy.

3.1 Image Preprocessing

The proposal in [Wang et al., 2021] collected 1065 CT images corresponding to 259 patients, 180 of which are typical viral pneumonia cases and 79 are COVID-19 confirmed cases. Regions of Interest (ROI) were manually outlined to these images to obtain the data that feeds the *deep learning* model:

- Converting the image to grayscale.
- Binarize the grayscale. The minimum frequency of all gray pixel frequency histograms is selected as the binarization threshold.
- Filling the background area. The *flood fill*[1] method is used to expand the image by one black pixel, and the black pixels near the edge are converted to white.

[1] An algorithm that determines and modifies the area connected to a given node in a multidimensional array with some matching attribute.

– Reverse the colors and determine that the two largest contours correspond to the lungs.

– The smaller bounding rectangle of the lung area is considered the ROI frame and the original image is cut out to obtain the final images.

To improve the reliability of the model, the ROIs are resized to a fixed size of $299 \times 299 \times 3$ pixels, copying the same image along each RGB channel, thus creating a virtual RGB image. They are separated into 3 sets:

– **Training**: 320 ROIs extracted from CTs belonging to *Xi'an Jiaotong University First Affiliated Hospital*, of which 160 correspond to typical viral pneumonia and 160 correspond to COVID-19.

– **Internal Validation**: 455 ROIs extracted from CTs belonging to *Xi'an Jiaotong University First Affiliated Hospital*, of which 360 correspond to typical viral pneumonia and 95 correspond to COVID-19.

– **External Validation**: 290 ROIs extracted from CTs belonging to *Nanchang University First Hospital* and *Xi'an No.8 Hospital of Xi'an Medical College*, of which 220 correspond to typical viral pneumonia and 70 correspond to COVID-19.

On the other hand, the proposal of [Song et al., 2021] collected CTs pertaining to 274 patients, of which 88 are COVID-19 positive cases, 100 are bacterial pneumonia cases, and 86 are healthy patients.

As with the method described above, a series of steps were followed by authors to obtain the data used to train the *deep learning* model, which are detailed below:

– CT images can contain more than 200 slices, so slices close to each other tend to be similar. To mitigate this, only 15 representative and equidistant slices are selected. This increases calculation speed and reduces the impact caused by the different number of slices in CTs from different hospitals

– Some slices contain incomplete lungs, the *OpenCV* library is used to detect these cases and those with lung regions occupying less than 50% of the total image are discarded.

– The contours of the lungs can vary substantially from person to person, and the *deep learning* model may over adjust the characteristics of these contours, therefore, the empty areas are filled with lung sections.

Finally, 777 slices of positive COVID-19 were obtained from CTs of *Renmin Hospital of Wuhan University* and *Third Affiliated Hospital*; 505 slices of bacterial pneumonia and 708 slices of healthy patients, both from CTs of *Renmin Hospital of Wuhan University* and *Sun Yat-Sen Memorial Hospital*.

Sets of training, validation and *tests* were created by a separation of $60\%/30\%/10\%$ respectively.

The proposal of [Song et al., 2021] does not detail the dataset used by them, however it is interesting what they propose and serves as a point of comparison with previous methods.

Since working with volumetric data, a special process is followed to prepare the images for model training. The steps are detailed below:

- A resampling, resizing and normalization of the CTs is performed.
- The resulting CTs are sampled, segmented using *fuzzy c means*[2] and features are extracted with the *gray-level co-occurrence matrix* method to obtain lung regions anomalies. To avoid morphological changes due to sampling, the same scaling factor is used in all dimensions and a zero *padding* is performed to obtain the final dimensions ($138 \times 256 \times 256$).
- Windows are selected within the CTs, scaled to increase contrast and extracted to obtain sub-volumes with the window dimensions.
- The resulting volumes are standardized.

3.2 *Deep Learning* Models

The proposal in [Wang et al., 2021] consists of an architecture composed of 3 main processes: CT preprocessing; ROIs *feature extraction* and training; classification with two dense layers and prediction using binary classifiers.

Transfer learning is performed using the CNN *Inception V3*, which is pre-trained with 1.2 million images from the *ImageNet* dataset. The resulting neuronal network is divided in two parts: the first part consists of a modified *Inception* network *(M-Inception)* and pre-trained to convert the images into one-dimensional feature vectors, whereas the second part consists of a dense neuronal network used for classification.

The difference between traditional *Inception* and *M-Inception* lies in the last dense layer, which was modified to reduce the dimensions of the *features*.

On the other hand, the proposal of [Song et al., 2021] consists of an architecture composed of 3 steps: CT preprocessing; obtaining slice-level predictions using the *Details Relation Extraction* network (*DRENet*); and the aggregation of the predictions to obtain a patient-level diagnoses.

The *DRENet* is built on the basis of the *ResNet-50* by adding a *Feature Pyramid Network* (FPN) to extract the details of each image. The FPN is modified to identify the regions with injuries of different sizes and an attention module is coupled to it to learn the importance of the features.

The flow that the *DRENet* is as described below:

- Each slice is entered into a *ResNet-50* to extract the *feature map*, emphasizing small regions, since lesions are generally small and the use of large *feature maps* only introduces noise to the model.
- The *feature map* goes through a *pooling* layer and a dense layer to obtain the global *features*. The FPN uses the latter to identify important regions within the image and assigns them a score.
- According to these regions, the top K sub-images are cut and at the same time, a new image is generated from the original one by multiplying the regions of the top K sub-images with their corresponding scores. The regions not present in the top K sub-images are set to zero.
- The *ResNet-50* is fed with the new image and the previous sub-images to extract local *features* within the latter and relational *features* between them.

[2] A form of clustering in which each datum can belong to more than one *cluster*.

– Both local and relational *features* are concatenated with global *features* and introduced into a multilayer perceptron to obtain the prediction at the cutoff level.

Predictions are made at the cutoff level for each patient and *mean pooling* is used to aggregate the resulting values, obtaining the prediction at a patient level.

The *deep learning* model is called *3D ImpCNN*. They do not go into details about the structure. However, some important aspects are highlighted:

– The weights of the dense layer are propagated to the *feature maps* of the convolutional layers to generate attention maps.

– The network stores the information in volumes through the application of *3D max pooling*.

– The *loss* function corresponds to a *binary cross entropy*.

3.3 Results

The *deep learning* model presented by [Wang et al., 2021] was trained for 15000 *epochs* using an initial learning rate of 0.01, automatically adjusted during the training. Only the modified part of the *M-Inception* is trained, and the *Adaptive Moment Estimation Gradient Descent* is used as an enhancer (optimizer). To improve the classification *accuracy*, an *ensemble*[3] of classifiers is used. The results are shown in Table 1.

Metric	Internal Validation	External Validation
AUC (95% CI)	0.93 (0.90 to 0.96)	0.81 (0.71 to 0.84)
Accuracy, %	89.5	79.3
Sensitivity	0.88	0.83
Specificity	0.87	0.67
PPV	0.71	0.55
NPV	0.95	0.90
F1-score	0.77	0.63

Table 1: Performance of the deep learning model to distinguish between typical viral pneumonia and COVID-19.

On the side of [Song et al., 2021] the number of sub-images to be extracted from the *inputs* in $K = 3$ was set. The model was trained in two different tasks: discriminate between healthy patients and patients with COVID-19 shown in Table 2; and distinguish between patients with COVID-19, patients with bacterial pneumonia and healthy patient shown in Table 3 (where the patient-level metrics were obtained when evaluating in the external validation set).

Finally, the results obtained by [Subramaniyan et al., 2021] in the task of discriminating between patients with positive COVID-19 and healthy patients are shown in Table 4.

[3] It consists of training more than one neural network on the same data set, then using each of the trained models to make a prediction before combining the forecasts to obtain a final prediction. This helps to reduce the variance of the predictions.

Data	AUC	*Accuracy*	*Precision*	*Recall*	*Specificity*	*F1 score*
Internal Val.	0.93	-	-	-	-	-
External Val.	0.95	86%	0.79	0.96	0.77	0.87

Table 2: Performance of the deep learning model to distinguish between COVID-19 and healthy.

Data	*Accuracy*	*Precision*	*Recall*	*Specificity*	*F1 score*
External Val.	93%	0.86	0.93	0.93	0.93

Table 3: Performance of the deep learning model to distinguish between COVID-19, bacterial pneumonia and healthy

COVID-19 classification on CT images of the thorax is based on the use of cross-sectional slices due to limitations such as training time, computational complexity inherent to the use of volumetric data and limited hardware resources. The advantage of working with volumetric data is that three dimensions of information are available, that are consolidated into spatial and temporal information.

To compensate for this loss, techniques such as image segmentation have been used to provide the neural networks with additional information about the images, as well as using methodologies such as separately analyzing the slices of a CT scan and weighting the results to obtain a global view of the diagnosis. The use of cross-sectional slices implies that the temporal dimension, or in this case, the height at which the lung lesions are found, is lost. This can be key when identifying anomalies, since the same lesion could be a symptom of different diseases depending on whether it is located in the upper or lower part of the lung.

The aim of this proposal is to develop a neural network model that takes as input the volumetric data from CT scans (3D) and is able to identify anomalies without requiring sophisticated data preprocessing techniques. In order to take advantage of all the tomography information, it is proposed to use a CNN composed of 3D layers, capable of applying convolution and pooling operations on the complete images. We propose the use of an *autoencoder* to extract key information from the 3D volumetric images and create a feature matrix to be used together with the main convolutional network, to increase the reliability of the classification.

In search of the solution, a first *ResNet-50* model was implemented, whose performance was lower than those presented in the literature, either because the size of the images is smaller than that used in the other studies or because the previous processing of the data they made is relevant.

The datasets used in different studies in the literature are different and the processes applied in some works are highly complex, so it is not easy to compare results with those in the literature.

In [Li et al., 2020] a 3D Deep Learning framework, called COVNet, was developed to extract visual features from volumetric chest CT scans for the detection of COVID-

Data	*Accuracy*	*Precision*	*Recall*	*Specificity*	*F1-score*
Validation	96.5%	96.1%	95.2%	96.2%	95.6

Table 4: The performance of the deep learning model to distinguish between COVID-19 and healthy

19. Diagnostic performance was assessed with the area under the receiver operating characteristic curve, sensitivity, and specificity. The sensitivity for detecting COVID-19 in the independent test set was 90% (95% confidence interval [CI]: 83%, 94%; 114 of 127 scans) and specificity 96% (95% CI: 93%, 98%; 294 of 307 scans), with an AUC of 0.96. The sensitivity for detecting community-acquired pneumonia in the independent test set was 87% (152 of 175 scans) and specificity 92% (239 of 259 scans), with an AUC of 0.95 (95% CI: 0.93, 0.97).

The algorithm shown in [Wang et al., 2020] obtained 0.959 ROC AUC and 0.976 PR AUC. When using a probability threshold of 0.5 to classify COVID-positive and COVID-negative, the algorithm obtained an accuracy of 0.901, a positive predictive value of 0.840 and a very high negative predictive value of 0.982.

In [Gaur et at., 2023] pre-trained CNN models are evaluated. Their results show that the proposed approach produced a high-quality model, with an overall accuracy of 92.93%, COVID-19, a sensitivity of 94.79%. This article shows an interesting comparative analysis of the study.

4 Proposed Solution

4.1 Dataset

The dataset used for training our proposed neural network corresponds to chest CT scans provided by Moscow municipal hospitals as part of the ***Molecular Sciences for Medicine*** [Morozov et al., 2020]. It consists of 1100 studies in **NifTI** format, of which 856 contain lesions attributed to COVID-19 and 254 are of healthy individuals or with lesions due to other diseases. On the other hand, CT scans store the intensity of *voxels* in *Hounsfield* units (HU); for this dataset, the intensities range from -1000 to 2000 HU.

The distribution of the people who participated in this study is:

- 42% men, 56% women and 2% did not report gender.
- Age ranged from 18 to 97 years with a medium of 47 years.

In order to use the dataset as part of the neural network training, it was necessary to perform a previous process consisting of three steps:

- Rotate the volumes by $90°$ to leave them in a fixed orientation.
- Standardize the intensity of the *voxels* so that the values are between 0 and 1. For this, -1000 HU is used as the lower level and $400HU$ as the upper level, the latter because above $400HU$ it is possible to find bones with different radio intensities that are not relevant for the application proposed in this article.
- Resize the volumes, leaving them with $128 \times 128 \times 64$ dimensions to reduce the necessary hardware resources.

4.2 Autoencoder

An *autoencoder* is an Artificial Neural Network capable of learning to compress and encode data efficiently, and to use this encoding to reconstruct a representation as accurate as possible of the original data.

The basic structure of an *autoencoder* is composed of four parts:

- ***Encoder***: Component responsible for the reduction of data dimensions and data compression to obtain an encoded representation.
- ***Bottleneck***: Layer containing the encoded representation of the data (feature vectors or compressed representation).
- ***Decoder***: Component that decodes the representation generated by the encoder to obtain a result as close as possible to the original.
- ***Reconstruction Loss***: Loss function used to measure the performance of the decoder and how closely the result resembles the original data.

The encoder and decoder may consist of dense, convolutional or LSTM layers, depending on the nature of the data and the application. We propose the use of an *autoencoder* with 3D convolutional layers, ***Mean Squared Error*** loss function and ***Adam*** optimizer, which has as input images of dimensions $128 \times 128 \times 64$ and generates a matrix of dimensions $16 \times 16 \times 8 \times 32$ extracted from the *encoder* (see Figure 2.)

It was trained for 100 *epochs*, with a batch of 16 images and a separation of $75\%/25\%$ for the training and validation set respectively.

To visualize its performance, the CT with a COVID-19 diagnosis of Figure 3(a) was taken, as an example, and the reconstruction obtained by the *autoencoder* is shown in Figure 3(b).

As can be seen, the reconstruction loses the fine details of the image, but the information on the general structure of the lungs is preserved. This may be relevant, since one of the problems of working with thorax CT scans is that neural networks tend to focus on the edges of the body rather than the interior of the lungs, requiring pre-processing of the images to mitigate this behavior.

4.3 ResNet

The *ResNets* are very deep neural networks that employ a type of interlayer connection to address the problems associated with using many layers (e.g., vanishing gradients). These special connections are called *skip connections*, and their function is to provide a layer with the *output* of a previous layer without passing through other intermediate layers. A set of layers with *skip connections* is known as residual block and is the basis of *ResNets*.

The proposed solution considers the use of a *ResNet* with 50 layers, including 3D convolutional layers. Unlike a *ResNet* used for classification, the *output* of this network is a vector of 512 *features* that will be used in combination with the vector generated by the *autoencoder* to perform the classification of CT images.

The architecture of this network can be divided into 6 parts:

- *Input* pre-processing.
- Block composed of 1 convolutional segment and 2 identity segments.
- Block composed of 1 convolutional segment and 3 identity segments.
- Block composed of 1 convolutional segment and 5 identity segments.
- Block composed of 1 convolutional segment and 2 identity segments.
- Block composed of dense layers.

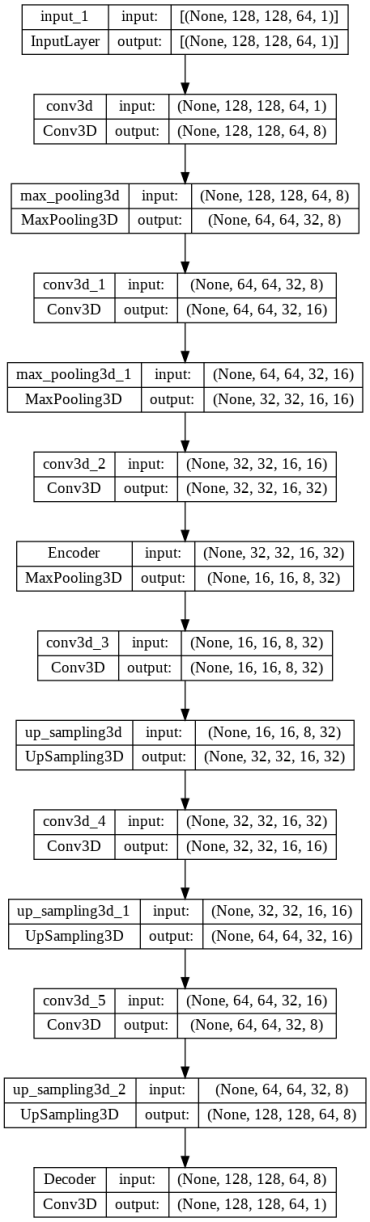
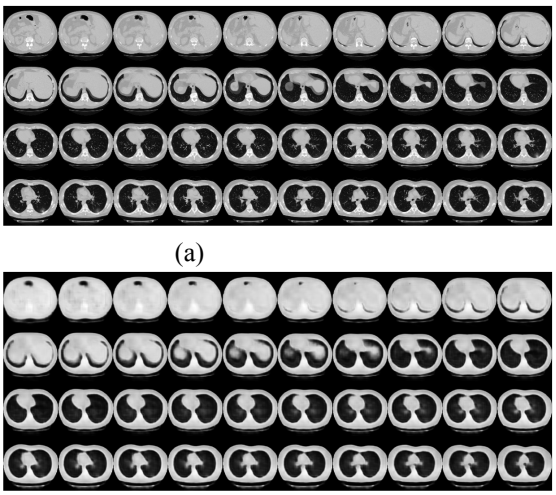


Figure 2: Convolutional autoencoder architecture with 3D layers. Source: Own.

4.4 Proposed Model

The proposed model is comprised of the already mentioned *ResNet* and a simple convolutional network responsible for interpreting the characteristics matrix generated by the



(a)

(b)

Figure 3: (a) CT thorax scan. (b) CT thorax reconstruction. Source: Own.

autoencoder. The outputs of both networks are concatenated and passed through dense layers to obtain the image classification.

The motivation behind our proposal comes from the assumption that the use of 3D images, while providing more information that can be captured by the neural networks, results in poorer performance due to the hardware resources needed to train the neural networks (smaller *batches* must be used during training, images have smaller dimensions, etc.). To solve this problem, extra information is added during the training of the main network, which is provided by the matrix generated by the *autoencoder*, containing the most relevant characteristics of the images in a compressed form.

Thus, a model composed of two concatenated neural networks is obtained, with ***Binary Cross Entropy*** loss function, and ***Adam*** Optimizer with exponential decay in the learning rate. The *input* is an array composed of the images measuring $128 \times 128 \times 64$ and the characteristics matrix generated by the *autoencoder* measuring $16 \times 16 \times 8 \times 32$.

254 images corresponding to COVID-19, and 254 images corresponding to Non-COVID-19 studies were selected, to avoid having an unbalanced dataset; these were separated by $60\%/20\%/20\%$ for a set of training, validation and *test* respectively, using stratified sampling, thus ensuring that all sets have the same proportion of categories. Training was performed for 85 *epochs*, with a *batch* of 8 images and taking a *checkpoint* of the iterations with the best *accuracy* over the validation set.

An extract of the complete architecture of the model is detailed in Figure 4, showing the concatenation with the additional component responsible for processing the characteristics matrix generated by the *autoencoder*.

A set of *tests* was used to assess the performance of the proposed solution, composed of 102 images corresponding to the following classes: 51 COVID-19 and 51 Non-COVID-19. The model is not exposed to this set during training, in order to obtain metrics free of bias and thus visualize the performance of the proposal.

Deep learning models deliver probabilities when making a prediction. Therefore,

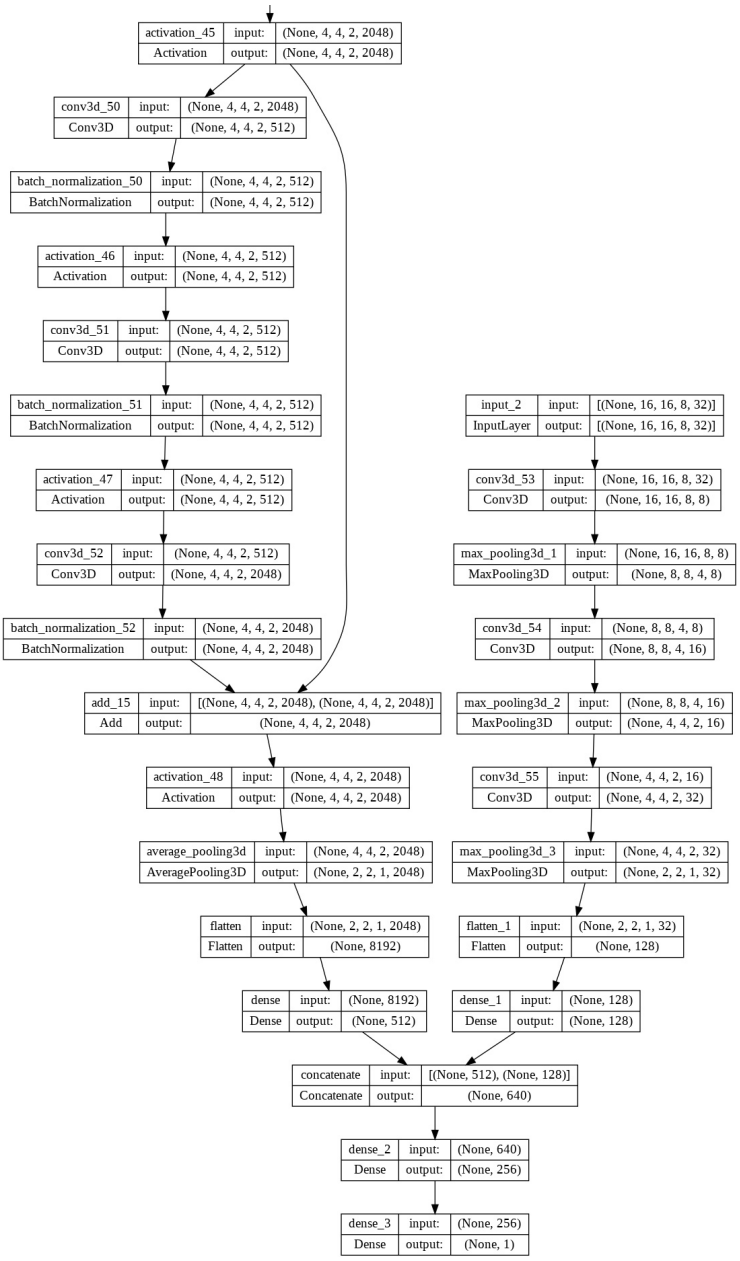


Figure 4: Proposed model: Concatenation of the two neural networks. Source: Own.

in order to categorize these probabilities among the different classes, it is necessary to determine a threshold at which a prediction is considered to be one class or another. In other words, the probabilities provided by the models are a measure of the confidence

that the model has on whether the *input* belongs to the positive class (COVID-19 in this case) or not. For this purpose, the threshold selected is 0.6, since it is a non-strict value and leaves less margin for the network to classify randomly.

Once the predictions are categorized, the confusion matrix is generated, as shown in Figure 5.

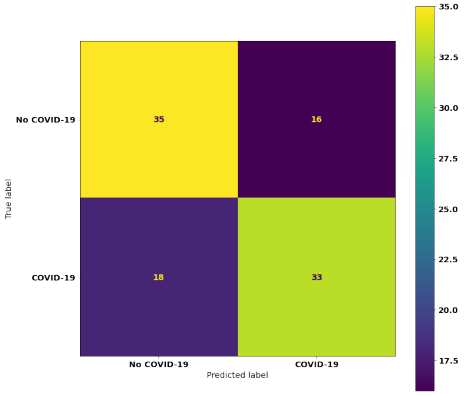


Figure 5: Confusion matrix of the proposal on the test set. Source: Own.

The diagonals of the confusion matrix indicate a coincidence between the predictions and the actual values, therefore, although a slight confusion between the classes is observed, in general the proposed model has the ability to discriminate between the diagnoses. It should be emphasized that the number of each diagnosis that was correctly classified is similar in both cases, so it can be said that the model does not have a preference or bias to one or the other.

To further analyze performance, metrics are obtained from the confusion matrix using the equations defined in the conceptual framework, shown in Table 5.

Data	AUC	*Accuracy*	*Precision*	*NPV*	*Recall*	*Specificity*	*F1-score*
Validation	0.70	70%	0.73	0.76	0.63	0.76	0.67
Test	0.67	67%	0.67	0.69	0.65	0.69	0.66

Table 5: The performance of the deep learning model to classify between COVID-19 and Non-COVID-19

The AUC represents the ability of the model to differentiate between the different classes, so values close to 1 indicate that the model can correctly classify the data into their corresponding classes. The proposed model obtained 0.67 on the *test* set, so it has some ability to properly differentiate the diagnoses.

On the other hand, the *accuracy* is a measure of the ratio between correct predictions and total predictions. It is usually not reliable in problems with unbalanced data sets, but in the case of this proposal, the same amount of data is available for both classes. Values close to 1 are better and the proposed model obtained an *accuracy* of 0.67 in the

independent set or in the *test*, so it can be said that 67% of the predictions corresponds to the real diagnosis.

The *precision* measures the proportion of positive predictions that were properly classified, i.e., predictions that were categorized as COVID-19 and that actually correspond to images with such a diagnosis. It is also used to assess the number of FP, whereby values close to 1 indicate that the model is able to correctly identify the positive categories and has few cases of FP in comparison. The proposed model achieved 0.67 in the set of *test*, so there are a certain number of FP or Non-COVID-19 diagnoses that were classified as COVID-19.

The *NPV* is the inverse of the *precision*, so it measures the percentage of negative predictions that were correctly classified, i.e., CT images that were classified as Non-COVID-19 and actually have that diagnosis. Values close to 1 are better, and indicate that the model is able to correctly identify the negative categories, presenting also a low number of FN, which is really important as it can be life-threatening for patients. The proposed model achieved 0.69 in the *test* set.

The *recall* is a measure of the proportion of positive cases that were correctly classified, i.e., images that have COVID-19 as a diagnosis and that the model was able to classify as such. Unlike *precision* and FP, *recall* takes into account the number of FN, so that values close to 1 indicate a lower number of FN. The proposed model achieved 0.65 in the *test* set, so there is a certain number of images corresponding to COVID-19 that were classified as Non-COVID-19.

The *specificity* is the opposite of *recall*; thus, it is a measure of the proportion of negative cases that were correctly classified. In the domain of this proposal it would correspond to the percentage of Non-COVID-19 diagnoses that were classified as such. Values close to 1 are better, the proposed model obtained 0.69 in the *test* set, so there is a number of Non-COVID-19 diagnoses that were classified as COVID-19, but still does not equal the amount that was correctly rated.

The *F1-score* is the harmonic average with *precision* and *recall*, so it provides a summary of how good these metrics are. Values close to 1 are better, and the proposed model scored 0.66 on the *test* set, a value that is reflected in the individual metrics listed above.

To validate the proposal, a model consisting of a classic *ResNet-50* is built, under the same training conditions but without the additional information provided by the feature matrix generated by the *autoencoder*. Figure 6 shows the confusion matrix obtained.

As can be appreciated, *ResNet-50* presents similar quantities of COVID-19 diagnoses that were classified as such, but were also categorized as Non-COVID-19. Therefore, it can be said that it has a greater bias towards the negative class, and it is better at identifying images with a Non-COVID-19 diagnosis.

Generally, the metrics obtained by the *ResNet-50* are similar to those of the proposed model, being the biggest improvement in the *recall*. This is well mirrored in the confusion matrix and as was earlier mentioned, in the confusion of the *ResNet-50* when classifying COVID-19. Similarly, the *specificity* of *ResNet-50* is higher, as it is better at identifying Non-COVID-19 cases.

Table 6 shows a comparison of the metrics of the proposed model (with *autoencoder*) and the *ResNet-50* without the *autoencoder* module on the *test* set.

Based on observations, it can be assumed that the information generated by the *autoencoder* is useful to improve the performance when classifying COVID-19. This is demonstrated when comparing the confusion matrices, where the proposed model has a more homogeneous behavior, without showing a preference for a particular diagnosis.

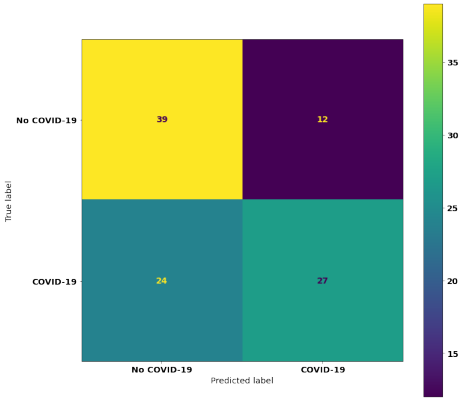


Figure 6: Confusion matrix when assessing a ResNet-50 in the test set. Source: Own.

Models	AUC	*Accuracy*	*Precision*	*NPV*	*Recall*	*Specificity*	*F1-score*
Proposal	0.67	67	0.67	0.69	0.65	0.69	0.66
ResNet-50	0.65	65	0.69	0.76	0.53	0.76	0.60

Table 6: Comparison between the proposal with autoencoder and the ResNet-50 on the test set to classify between COVID-19 and Non-COVID-19.

5 Conclusions and Future Work

The proposed solution aims to build a *deep learning* model that uses 3D CT images as *input* and classifies them into COVID-19 or Non-COVID-19. The differentiating factor of the proposal is that it works with 3D images and not with cross-sectional slices (2D), as is common to see in other state-of-the-art proposals. In addition, the images do not need to be subjected to particularly complex processing prior to the analysis. Based on this, it was possible to develop a model that has great potential of improvement. This potential lies in the modular nature of the solution, since it consists of a baseline architecture to which components can be added thus providing additional information about the images. In this case, an *autoencoder* was used as a way of obtaining relevant features that support the main architecture to understand the finer details of the images and therefore, be able to discriminate between COVID-19 and Non-COVID-19 more reliably.

A restriction in the dimensions of the CT images used by the proposed model is that their dimension must be $128 \times 128 \times 64$, which adds an additional restriction to the type of studies that can be analyzed, especially if they have less than 64 slices.

Generally, metrics obtained when assessing the proposed model are encouraging, the most important being AUC with a value of 0.67, *recall* with 0.65 and *specificity* with 0.69. These metrics indicate that the model is able to distinguish between a COVID-19 image with a Non-COVID-19 image, and can accurately distinguish each class most of the times, and shows no tendency to choose one over the other.

One challenge to be faced in future works is the improvement of performance. In some sense, a way to achieve a better performance is to discard some intermediate cuts, since changes between consecutive cuts is irrelevant and introduces noise. Other important think is that exams must be validated by experts to ensure they are well labeled,

avoiding a potential source of errors. Both models (with and without the *autoencoder* module) had access to the complete exams with minimal alterations. The classification was not carried out at the cut-off level nor was an aggregation of the classifications of each cut-off made to obtain a result at the exam level, this being a source of potential and important improvement.

The proposed solution can categorize into COVID-19 and Non-COVID-19, but as a future work the proposal could be modified to facilitate the diagnosis of different lung diseases limited only by the dataset used.

Acknowledgements

Thanks to IDEA FONDEF fund under grant IT21I0019.

References

[Adams et al., 2020] Adams, H., Kwee, T., Yakar, D., Hope, M., Kwee, R.: "Chest CT Imaging Signature of Coronavirus Disease 2019 Infection: In Pursuit of the Scientific Evidence". Chest, 158, 5, (2020), 1885–1895, Elsevier.

[Akdag et al., 2022] Akdag, S., Kuncan, F. and Kaya, Y.: (2022) "A new approach for congestive heart failure and arrhythmia classification using downsampling local binary patterns with LSTM". Turkish Journal of Electrical Engineering and Computer Sciences, 30, 6, Article 10, (2022). https://doi.org/10.55730/1300-0632.3930.

[Badr, 2019] Badr, W.: "Auto-Encoder: What Is It? And What Is It Used For". https://towardsdatascience.com

[Cengil and Cinar, 2018] Cengil, E. and Çinar, A.: "A deep learning based approach to lung cancer identification". Int. conf. on artificial intelligence and data processing (IDAP), Malatya, Turkey, pp. 1–5. (2018). https://doi.org/10.1109/IDAP.2018.8620723

[Dansana et al., 2023] Dansana, D., Kumar, R., Bhattacharjee, A. et al.: "Early diagnosis of COVID-19-affected patients based on X-ray and computed tomography images using deep learning algorithm"; Soft Comput 27, 2635–2643 (2023). https://doi.org/10.1007/s00500-020-05275-y

[Eken, 2023] Eken, S. (Retracted Article): "A topic-based hierarchical publishsubscribe messaging middleware for COVID–19 detection in X–ray image and its metadata", Soft Comput 27, 2645–2655 (2023). https://doi.org/10.1007/s00500-020-05387-5

[Gaur et at., 2023] Gaur, L., Bhatia, U., Jhanjhi, N.Z. et al.: "Medical image-based detection of COVID-19 using Deep Convolution Neural Networks"; Multimedia Systems 29, 1729–1738 (2023). https://doi.org/10.1007/s00530-021-00794-6

[Kaya et al., 2022a] Kaya, Y., Kuncan, F. and Ertunç, H. M.: "A new automatic bearing fault size diagnosis using time-frequency images of CWT and deep transfer learning methods". Turkish Journal of Electrical Engineering and Computer Sciences: 30, 5, Article 12, (2022). https://doi.org/10.55730/1300-0632.3909

[Kaya et al., 2022b] Kaya, Y., Kuncan, F. and Tekin, R.: "A New Approach for Congestive Heart Failure and Arrhythmia Classification Using Angle Transformation with LSTM", Arab J Sci Eng 47, 10497–10513 (2022). https://doi.org/10.1007/s13369-022-06617-8

[Kaya et al., 2022c] Kaya, Y., Yiner, Z., Kaya, M. and Kuncan, F.: "A new approach to COVID-19 detection from x-ray images using angle transformation with GoogleNet and LSTM". Meas. Sci. Technol. 33 (2022) 124011 (14pp). https://doi.org/10.1088/1361-6501/ac8ca4

[Kwee and Kwee, 2020] Kwee, T. and Kwee, R.: "Chest CT in COVID-19: What the radiologist needs to know"; RadioGraphics, 40, 7, (2020), 1848–1865.

[Li et al., 2020] Li, L., Qin, L., Xu, Z., Yin, Y., Wang, X. et al.: "Artificial Intelligence Distinguishes COVID–19 from Community Acquired Pneumonia on Chest CT"; Radiology; 296:E65–E71, (2020). 10.1148/radiol.2020200905

[Morozov et al., 2020] Morozov, S., Andreychenko, A., Pavlov, N., Vladzymyrskyy, A., Ledikhova, N., Gombolevskiy, V., Blokhin, I., Gelezhe, P., Gonchar, A. and Chernina, V.: "Mosmeddata: Chest CT scans with covid–19 related findings dataset", arXiv preprint arXiv:2005.06465. (2020).

[Nair et al., 2021] Nair, R., Alhudhaif, A., Koundal, D., Doewes, R. I. and Sharma, P.: "Deep learning-based COVID-19 detection system using pulmonary CT scans"; Turkish Journal of Electrical Engineering and Computer Sciences: 29, 8, Article 9, (2021). https://doi.org/10.3906/elk-2105-243.

[Pianykh, 2009] Pianykh, O.: "Digital imaging and communications in medicine (DICOM): a practical introduction and survival guide"; Springer Science & Business Media. (2009).

[Rubin et al., 2020] Rubin, G., Ryerson, C., Haramati, L., Sverzellati, N., Kanne, J., Raoof, S., Schluger, N., Volpi, A., Yim, J-J, Martin, I. and others: "The role of chest imaging in patient management during the COVID–19 pandemic: a multinational consensus statement from the Fleischner Society"; Chest, Elsevier. (2020).

[Schaller et al., 2020] Schaller, T., Hirschbühl, K., Burkhardt, K., Braun, G., Trepel, M., Martin, M., Märkl, B., Claus, R.: "Postmortem examination of patients with COVID-19". JAMA, (2020).

[Sethy et al., 2020] Sethy, P.K., Behera, S.K., Ratha, P.K., Biswas, P.: "Detection of coronavirus disease (COVID–19) based on deep features and Support Vector Machine"; International Journal of Mathematical, Engineering and Management Sciences. 5 (4), 643–651, (2020). https://doi.org/10.33889/IJMEMS.2020.5.4.052

[Shah et al., 2020] Shah, V., Keniya, R., Shridharani, A., Punjabi, M., Shah, J. et al.: "Diagnosis of COVID–19 using CT scan images and deep learning techniques"; Emergency Radiology; 28: 497–505, (2020). 10.1007/s10140-020-01886-y

[Song et al., 2021] Song, Y., Zheng, S., Li, L., Zhang, X., Zhang, X., Huang, Z., Chen, J., Wang, R., Zhao, H., Zha, Y. and others: "Deep learning enables accurate diagnosis of novel coronavirus (COVID–19) with CT images"; IEEE/ACM Transactions on Computational Biology and Bioinformatics. (2021).

[Subramaniyan et al., 2021] Subramaniyan, M., Sampathkumar, A., Jain, D., Ramachandran, M., Patan, R., Kumar, A.: "Deep Learning Approach Using 3D-ImpCNN Classification for Coronavirus Disease"; Artificial Intelligence and Machine Learning for COVID–19, pp. 141–152, Springer, (2021).

[Wang et al., 2020] Wang, X., Deng, X., Fu, Q., Zhou, Q., Feng, J., Ma, H., Liu, W. and Zheng C.: "A Weakly–Supervised Framework for COVID–19 Classification and Lesion Localization From Chest CT"; IEEE Transactions on Medical Imaging, 39 (8), August (2020), 2615–2625. 10.1109/TMI.2020.2995965

[Wang et al., 2021] Wang, S., Kang, B., Ma, J., Zeng, X., Xiao, M., Guo, J., Cai, M., Yang, J., Li, Y., Meng, X. and others: "A deep learning algorithm using CT images to screen for Corona Virus Disease (COVID-19)"; European radiology, pp. 1–9, Springer, (2021),

[WHO, 2020] WHO: "Coronavirus disease (COVID-19) pandemic"; https://www.who.int/emergencies/diseases/novel-coronavirus-2019, (2020). Accessed on: Nov 26, 2022.

[Yilmaz, 2021] Yilmaz, A.: "Diagnosing COVID-19 from X-Ray images with using multichannel CNN architecture"; Journal of the Faculty of Engineering and Architecture of Gazi University, 36(4):1761-1773, (2021). (In Turkish).