COVID-19 Mortality Risk Prediction Using X-Ray Images

J. Prada[1], Y. Gala[1], A. L. Sierra[2] *

[1] Universidad Autónoma de Madrid, Cantoblanco, Madrid (Spain)
[2] Universidad Complutense de Madrid, Madrid (Spain)

Received 24 May 2020 | Accepted 16 February 2021 | Published 8 April 2021

unir
LA UNIVERSIDAD EN INTERNET

Abstract

The pandemic caused by coronavirus COVID-19 has already had a massive impact in our societies in terms of health, economy, and social distress. One of the most common symptoms caused by COVID-19 are lung problems like pneumonia, which can be detected using X-ray images. On the other hand, the popularity of Machine Learning models has grown exponentially in recent years and Deep Learning techniques have become the state-of-the-art for image classification tasks and is widely used in the healthcare sector nowadays as support for clinical decisions. This research aims to build a prediction model based on Machine Learning, including Deep Learning, techniques to predict the mortality risk of a particular patient given an X-ray and some basic demographic data. Keeping this in mind, this paper has three goals. First, we use Deep Learning models to predict the mortality risk of a patient based on this patient X-ray images. For this purpose, we apply Convolutional Neural Networks as well as Transfer Learning techniques to mitigate the effect of the reduced amount of COVID19 data available. Second, we propose to combine the prediction of this Convolutional Neural Network with other patient data, like gender and age, as input features of a final Machine Learning model, that will act as second and final layer. This second model layer will aim to improve the goodness of fit and prediction power of our first layer. Finally, and in accordance with the principle of reproducible research, the data used for the experiments is publicly available and we make the implementations developed easily accessible via public repositories. Experiments over a real dataset of COVID-19 patients yield high AUROC values and show our two-layer framework to obtain better results than a single Convolutional Neural Network (CNN) model, achieving close to perfect classification.

Keywords

Convolution Neural Network, Coronavirus COVID-19, Deep Learning, Machine Learning, Medical Images.

DOI: 10.9781/ijimai.2021.04.001

I. Introduction

MACHINE Learning (ML) [1], is a branch of Artificial Intelligence whose objective is to build systems that automatically learn from data. The popularity of ML techniques has grown exponentially in recent years and they have been applied to solve a wide variety of problems, such as stock market prediction [2], fraud detection [3], or renewable energy prediction [4], [5].

Although often considered an independent field, Deep Learning (DL) [6], is not less and not more than just another family of Machine Learning models. However, it is a family of models with some extremely relevant properties, such as its high predictive power and its ability to perform end-to-end learning. A specific family of Deep Learning techniques, called Convolutional Neural Networks (CNNs) [7], presents a set of properties highly advantageous for its use in image classification tasks and has in recent years become the state-of-the art for this type of problems.

Image recognition or image classification problems [8], are a set of tasks among the supervised learning [9] branch of ML problems which goal is to correct segment images into a pre-defined set of possible groups or classes. For instance, we may want to classify if an image contains a car, label 1, or not, label 0. Image classifications tasks show up often in the healthcare sector. Some examples of these problems will be Diabetic Retinopathy diagnosis [10], histological analysis [11], or tumor early detection [12].

Taking this into account, the aim and motivation of this research is to apply these techniques to predict the mortality risk of a COVID-19 patient using X-ray images and demographic data of the patient.

We divided our research in two different phases. The first step of this research is to use CNN models to predict the targeted mortality risk using solely X-ray images as input. We will call this model COVID-CheXNet.

Once this COVID-CheXNet model is built, we aim to train a second model, which will act as a second layer, which will use as input the output of our COVID-CheXNet, numeric information regarding characteristics of the X-ray image, and other basic demographic patient data like gender and age of the patient. For this purpose, we tested some of the most popular and powerful Machine Learning models like Neural Nets [13], Support Vector Machines (SVMs) [14] or Extreme Gradient Boosting (XGBoost) [15], together with Logistic Regression and Random Forest [16] models that will act as benchmarks.

* Corresponding author.
E-mail addresses: jesus.prada@estudiante.uam.es (J. Prada), yvonne.gala@estudiante.uam.es (Y. Gala), analusie@ucm.es (A. L. Sierra).

To test the usefulness of this new framework, experiments using a public dataset of COVID-19 X-ray image data collection are carried out. One of the main difficulties to build these models, often found in healthcare real problems, is the reduced amount of X-ray data available right now for COVID-19 patients, even more reduced when we add to this the necessity of knowing if the outcome of that patient was or not an Exitus. Transfer Learning [17] has shown to be a good method to mitigate the negative effects of this lack of data and will be the approach followed in this paper to try to solve this issue.

Theoretical details and code implementations for this two-layer framework, are developed and made publicly available, as well as datasets used in the experiments.

The novelty of our research is mainly due to two factors. The first one is the aim itself, as to our knowledge this is the first study that tries to predict the mortality risk of a COVID-19 patient using ML models based on X-ray images. The other main novelty factor is our proposed two-layer framework that allow us to combine a CNN prediction based on X-ray images with other numerical sources of information like demographical data of the patient, as past research about using X-ray images to make predictions about other lung diseases has focused solely on the use of a single CNN model.

The rest of this paper is organized as follows. In Section II we compare the motivation and limitation of related works. A brief review of prior theoretical background for the main ML models tested, Deep Learning and CNN basic concepts is presented in Section III. Section IV gives an in-depth description of the proposed method, both COVID-CheXNet layer and the final second ML layer, as well as implementation details. In Section V we describe experiments over a real-world public COVID-19 dataset and show the corresponding results. Section VI analyzes the results obtained in these experiments. Finally, the paper ends with the Section VII on conclusions and possible lines of future work.

II. Related Work

COVID-19 research publications based on the use of ML techniques are still limited, but some works have some common ground with our research.

CNN models have already been shown to achieve good performance when solving the image recognition problem of classifying if a patient have pneumonia or other lung related diseases based on X-ray images [18]. However, the aim here is different to the more specific task we want to tackle in our research, which is to completely focus only on the COVID-19 disease among all lung related health problems.

Convolutional Neural Networks have also been used to diagnose COVID-19 in patients based on X-ray images [19],[20] or CT scans [21]. However, we aim here to go a significant extra step and predict the mortality risk of this patient. We consider this to be much more helpful for clinicians, as when capable of performing an X-ray scan on a patient, clinicians will in most cases also be able to conduct a test for more accurate COVID-19 diagnosis, tests that are moreover getting cheaper and quicker to analyze with the passage of time.

Furthermore, these related studies directly use CNN models that use as input solely X-ray images. We propose here to combine this CNN predictions with a second layer model that also uses as input other numeric data, like demographical data about the patient and characteristics of the image. This is a critical difference as results show this two-layer framework greatly decreases prediction errors compared with the single CNN layer that only uses X-ray images as input.

Novelty of our approach is confirmed in [22], a recent paper that reviews research of AI applied for fighting coronavirus and that heavily mentions the use of DL techniques to diagnose COVID-19 but does not make any reference that points to the existence to this day of a research about the use of ML to predict mortality risk in these patients.

III. Prior Theoretical Background

A. Support Vector Machine

The aim of SVM is to obtain the best separating hyperplane possible between two o several different classes. We will focus here on the 2-class or binary problem. In real-world problems, usually finding a hyperplane which separates perfectly the data is not possible. Therefore, defining the slack variables $\xi = (\xi_1, \xi_2 \dots \xi_N)$, one natural way to define this problem will be

$$\begin{aligned} \max_{\beta,\beta_0} \quad & M \\ subject\ to \quad & y_i(x_i^T\beta + \beta_0) \geq M - \xi_i, i = 1,\dots,N \\ & ||\beta|| = 1 \end{aligned} \tag{1}$$

where M is the margin between the training points for class 1 and -1, ξ_i is the absolute value of the amount by which the prediction $f(x_i) = x_i^T\beta + \beta_0$ is on the wrong side of its margin.

Reference [23] shows that this problem is equivalent to the following convex constrained optimization problem

$$\begin{aligned} \max_{\beta,\beta_0} \quad & \frac{1}{2}||\beta||^2 + C\sum_{i=1}^N \xi_i \\ subject\ to \quad & y_i(x_i^T\beta + \beta_0) \geq 1 - \xi_i, i = 1,\dots,N \\ & \xi_i \geq 0, i = 1,\dots,N \end{aligned} \tag{2}$$

where the parameter C is often called *cost*. It is easy to see that the hard margin case corresponds to $C = 1$, that leads to $\Sigma\xi_i = 0$, i.e. not a single point on the wrong side of the margin.

The problem solved in practice is the dual formulation derived using Lagrangian techniques [24].

$$\begin{aligned} \max_{\alpha_i} \quad & L_D(\alpha_i) = \sum_{i=1}^N \alpha_i - \frac{1}{2}\sum_{i=1}^N \sum_{j=1}^N \alpha_i\alpha_j y_i y_j x_i^T x_j \\ subject\ to \quad & y_i(x_i^T\beta + \beta_0) \geq 1 - \xi_i, i = 1,\dots,N \\ & \xi_i \geq 0, i = 1,\dots,N \\ & \alpha_i \geq 0, i = 1,\dots,N \\ & \mu_i \geq 0, i = 1,\dots,N, \Rightarrow \alpha_i \leq C, i = 1,\dots,N \\ & \beta = \sum_{i=1}^N \alpha_i y_i x_i \\ & \sum_{i=1}^N \alpha_i y_i = 0 \\ & \alpha_i[y_i(x_i^T\beta + \beta_0) - (1 - \xi_i)] = 0. \\ & \mu_i\xi_i = 0, i = 1,\dots,N \Rightarrow (C - \alpha_i)\xi_i = 0, i = 1,\dots,N \end{aligned} \tag{3}$$

With the called Karush-Kuhn-Tucker conditions as restrictions.

Finally, using the kernel trick and a kernel function, $k(x_i; x_j)$, satisfying *Mercer's condition* [25] we can get the following analogous formulation

$$\max_{\alpha_i} \quad L_D = \sum_{i=1}^N \alpha_i - \frac{1}{2}\sum_{i=1}^N \sum_{j=1}^N \alpha_i\alpha_j y_i y_j K(x_i, x_j) \tag{4}$$

that allow us to extend the previous linear version of the SVM problem to a non-linear one.

B. Extreme Gradient Boosting

Boosting models aim to combine different individual models, usually called weak learners, into a single final more powerful model, commonly called strong learner.

In Boosting, weak learners are of a homogenous nature, i.e., they all come from the same family of models. Normally this family of models are decision trees or combination of them, Random Forest models.

These weak learners are trained in a sequential fashion. The basic idea is that each individual model would be a simple high bias model, like a shallow tree, and the subsequent weak learner will correct its errors, reducing the bias and increasing the goodness of the final model or strong learner.

In computational terms, the sequential nature of the method could be a drawback, but the aim is that this negative factor gets balanced by the fact that each individual weak learner is a basic low variance model and thus fast to train.

In Gradient Boosting models, the strong learner, S, is defined by the following equation

$$S = \sum_{i=1}^{L} c_i I_i \tag{5}$$

where I_i represents each one of the individual weak learners and c_i their corresponding coefficients.

In summary, Gradient Boosting follows this iterative algorithm:

1. Errors, E, are initialized with the target value to be predicted. Therefore, the first weak learner will predict the desired label.
2. Another individual model that predicts errors E is trained.
3. The new individual model is added to our final combined model, with c_i the coefficient that minimizes the global error of the new combined model S_k.
4. The value of the errors $E = E\ (S\_\ (k))$ corresponding the new combined model S_k is updated.
5. Steps 2-4 are repeated until the model converges or the maximum number of iterations is reached.

Extreme Gradient Boosting (XGBoost) is just an optimized implementation of standard Gradient Boosting models.

C. Artificial Neural Net

An Artificial Neural Net (ANN) model is made up of a collection of connected units called neurons, where the output of each neuron is computed by some non-linear function, called *activation function*, of the sum of its inputs. Neuron connections have weights, so activations of different neurons can have bigger impact than others. Neurons of one layer connect to neurons of the preceding and following layers. In between the input and output layers are zero or more hidden layers.

Given a training sample and a target to predict, an ANN will compute all the activation functions from the input layer to the output layer, obtaining a final prediction as a result. We call this a *forward* pass.

Once this forward pass has been performed, we need an algorithm to propagate backwards the error from the units in the output layer to the units in preceding layers to update model weights using techniques like gradient descent. This is called the *backward pass*. This algorithm is called backpropagation and is used to optimize ANNs. The goal of backpropagation is to be able to extend gradient descent to all the layers in the network. Backpropagation defines the error associated to a hidden unit as the weighted average of the errors of the units in the adjacent layer. The gradient descent for a layer j, with k as the next layer and i as the previous one, will have the following formulation

$$\frac{\partial E_L}{\partial w_{ji}} = \frac{\partial E_L}{\partial s_j}\frac{\partial s_j}{\partial w_{ji}} = \delta_j \frac{\partial s_j}{\partial w_{ji}} \tag{6}$$

where E_L represents the local error, w_{ji} is the weight of the connection from unit i to unit j, $s_j = \int w_{ji} z_i$ the sum of the weighted inputs of unit j, z_i the output of unit i, and δ_i the generalized error at unit j.

This can be shown [26] to be equivalent to

$$\frac{\partial E_L}{\partial w_{ji}} = (\int \delta_k w_{kj}) F'_j(s_j)\, z_i \tag{7}$$

D. Deep Artificial Neural Net

The concept of Deep Learning has had different interpretations in recent years. Deep learning is often employed simply to refer to a specific subset of Artificial Neural Networks. It is used to name ANNs with many hidden layers. However, the Deep Learning denomination has also been used to refer to any type of Machine Learning model framework which consists of an iterative process of several optimization steps or layers. An example of this is Deep Belief Networks (DBNs) [27], a type of ML models used for unsupervised learning. Another example of Deep Learning structure using models other than Neural Networks can be found in [28].

Nevertheless, it is true that clearly the link between Deep Learning and Deep Artificial Neural Nets is strong and almost ever-present nowadays. Several factors have probably had an impact on this, including the fact that ANNs schema adapts almost perfectly to the concept of DL framework and some of the first groundbreaking advances in DL corresponding to deep ANNs.

In recent years, the popularity of DL models has increased in a spectacular manner, due to the wide availability of powerful computing facilities, advances on the theoretical underpinnings of multilayer perceptrons (MLPs), several improvements on their training procedures and a better understanding of the difficulties related to many layered architectures, like better weight initialization methods and new activation functions such as Rectified Linear Unit (ReLU). To all these factors we can add the appearance of multiple development frameworks such as TensorFlow [29] and Keras [30].

E. Convolutional Neural Network

In the past, image classification Machine Learning models used raw pixels to classify the images. You can classify dogs for instance based on color histograms and edge detection, i.e. by color and ear shape. This method has been successful but has its limitations, especially when it encounters images with more complex patterns.

Convolutional Neural Networks are a type of neural network model which allows us to extract higher representations from an image. Unlike the classical image recognition where the image features are defined manually as a previous step, CNN takes the image's raw pixel data, trains the model, then extracts the features automatically for better classification.

This type of approach, where expert knowledge to pre-process the image is not needed, is usually known as *end-to-end learning*, and is one of the main reasons behind the recent popularity of these models.

In its most basic version, CNNs are a combination of two type of layers:

- Convolution layer: sweeps a moving window through images and then calculates the filter dot product of the pixel values. This allows convolution to emphasize relevant features.
- Pooling layer: Replaces output of convolution with a summary to reduce data size and processing time. This summary can be for instance the maximum or mean value among a set of several values. This allows pooling to determine features that produce the highest impact and reduces the risk of overfitting.

F. Transfer Learning

Until recently, conventional ML and Deep Learning algorithms have been traditionally designed to work in isolation. These algorithms are trained to solve a specific task and the models must be rebuilt from scratch once the task changes.

However, it is well-known that humans have an inherent ability to transfer knowledge from one task to another. What we acquire as knowledge while learning about one task, we can utilize in the same

way to solve related tasks. The more related the tasks, the easier it is for humans to transfer our knowledge.

Transfer Learning method tries to apply this same intuition to Deep Learning models, overcoming the isolated learning paradigm and utilizing knowledge acquired for one task to solve related ones.

The idea is that, when trying to solve a task using DL models, instead of training the model from scratch one can reutilize totally or partially other DL models trained to solve similar tasks. For instance, a model built to detect cats, could be reused to detect instead dogs.

There are four main transfer learning approaches, depending on how much reutilization of the previous model is done:

1. Reutilize only the Deep Learning structure, i.e., the configuration and order of the different layers. All the corresponding weights are trained from scratch using the data related to the new task.
2. Reuse the DL structure and use trained weights as initial values. All the weights will be updated using the new data, in a process usually called *fine tuning*.
3. Reuse the DL structure and the weights of some layers, update the rest. You will select a threshold layer, up until this layer all weights will remain fixed, the layer from this point to the output layer will be updated using the new data.
4. Reutilize the DL structure with the same weights. Model weights will not be adapted to the new task and only extra layers added to the base ones will serve to adapt the model to your task. This can only be a valid option when the two problems are similar.

IV. Proposed Method

This section aims to describe the technical details of the proposed ML framework to solve the task of predicting mortality risk for a COVID-19 patient. Details of the dataset and experiments carried out to test its efficacy are detailed in Section V.

A. First Layer

As described in Section I, the aim of our first layer is to build a model able to give a mortality risk using as input only X-ray images from COVID-19 patients, which we will call COVID-19 CheXNet. We decided that for this purpose the most suited family of models were CNN models, as they have proved repeatedly to be the best option in image classification tasks like the one in hand.

As stated before, one of the main difficulties when trying to solve our task was the lack of available data. Due to its novelty, there are not many X-ray images publicly available for patients with confirmed COVID-19 diagnosis. Furthermore, this shortage of availability was multiplied by the fact that in our case the target is the outcome, Exitus or no Exitus, of the patient. Datasets with both X-ray images and patient outcome were difficult to find and their volume small.

To deal with this drawback, we applied two methods: First, we make use of transfer learning techniques to take advantage of the knowledge extracted by CNN models from previous research in similar tasks. Second, we also applied data augmentation methods to create new synthetical X-ray images.

We describe our data augmentation approach in Section V.C, so we will focus here on the transfer learning methodology applied. CheXNet model [31] is a Convolutional Neural Network that achieves Radiologist-Level Pneumonia Detection on Chest X-Rays. It has been shown to have a margin of >0.05 AUROC over previous state of the art results and an F1 score of 0.435 (95% CI 0.387, 0.481), higher than the radiologist average of 0.387 (95% CI 0.330, 0.442). This CheXNet model is trained using a Deep Learning structure called Densenet-121, a 121-layer convolutional neural network, the simplest DenseNet among those designed over the ImageNet dataset. The Densenet-121 structure is shown in Fig. 1.

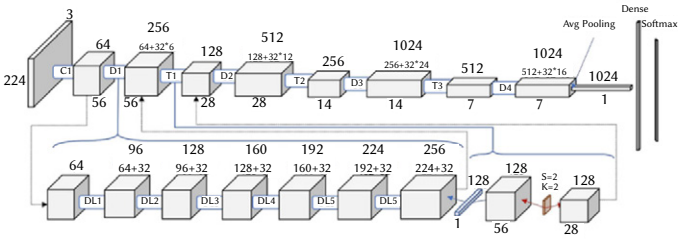

Fig. 1. Densenet-121 layer structure.

We use as our base model this CheXNet CNN. We opted to go for method 4 of transfer learning, as described in Section III.F, i.e., reusing the Densenet-121 structure and preserving the weights of CheXNet, fine-tuning only some additional layer weights to our new COVID-19 dataset.

To the Densenet-121 structure, we added two dense ReLU activation layers with 512 and 256 units, respectively. Finally, we added a logistic layer with sigmoid activation that will generate the final prediction of our model. This is binary classification problem, so only one unit is needed. All these layers are separated by dropout layers.

As our tackled problem represents an example of unbalanced classification task, i.e. there are more cases of non-Exitus label than Exitus outcomes, we set different class weights to balance the impact of each class on the CNN loss function. Therefore, errors in the minority class are penalized more than errors in the majority class.

All weights from the base CheXNet are frozen, i.e. not updated using our new data. Weights from these extra layers will define the correct adaptation of our COVID-19 CheXNet model to the problem we want to tackle.

Implementation of our proposed COVID-19 CheXNet in Python can be found on GitHub[1]. This implementation is based on the use of Keras.

B. Second Layer

Once we have a mortality risk prediction based solely on X-ray images coming from our first layer CheXNet model, the goal of our second layer model is to combine this prediction output with basic demographic data like gender, age and location, and basic details of the X-ray scan like the view used and the offset, to compute a new mortality prediction. This way we aim to get an improved mortality risk prediction with respect to the one obtained in the first layer, as we are now basing our prediction on additional information.

This is done using the following approach. Mortality risk prediction of layer 1 model becomes the first input column of a new input dataset, that has as remaining columns or input variables information related to demographics and X-ray image characteristics of each patient. As target of this dataset, we will use again the outcome of the patient, Exitus (1) or survival (0). This new combined dataset is passed as input to our second layer ML model to generate new and improved mortality risk predictions as our final output. The total list of input variables used as inputs of this second layer can be found in Table I.

To decide which model to use in this second layer we compute a grid search testing Logistic Regression, Random Forests, SVM, XGBoost and ANN models. The first two are more basic ML families, but we decided to include them due to having a low dimensionality dataset and to at least provide a good benchmark reference.

[1] https://github.com/jesuspradaalonso/COVID-19-CheXNet-

TABLE I. Input Data of Second Layer Model

Type	Variable
CheXNet	Mortality risk prediction
Demographics	Gender
	Age
	Location
X-ray	View
	Offset

We carry out hyperparameter optimization for each one of these families of models, as will be described in Section V.D.

The implementation needed to build this second layer model is also available on our GitHub[1], both in R and Python versions.

C. Two-layer Framework Diagram

Object process diagram of this two-layer framework is presented in Fig. 2.

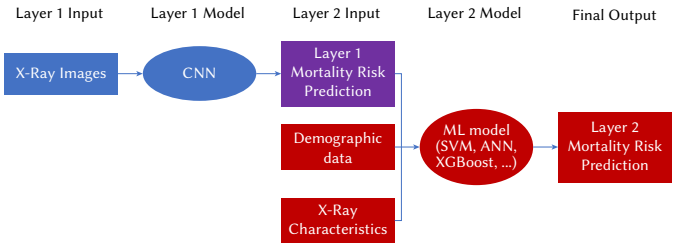

Fig. 2. Proposed two-layer framework diagram.

V. Experiments and Results

To test the performance of our proposed models described in Section IV we evaluate its goodness over an experiment based on public data available on COVID-19 patients.

A. Dataset

We used two sources to build our dataset:

- covid-chestxray-dataset[2]: GitHub with information, both X-ray images and basic clinical data, for 209 COVID-19 patients.
- Spanish society of medical radiology, SERAM, COVID-19 data[3]. From this source 12 registers where manually extracted.

Therefore, the combined dataset contains 221 registers. For each register the following information of the patient is available:

- X-ray chest image.
- Gender.
- Age.
- Hospital location.
- X-ray view: anteroposterior (AP) or posteroanterior (PA).
- X-ray offset.

This dataset can be found in our public GitHub repository[1].

B. Train/val/test Split

Although the optimal ratio of data used in train, validation and test depends on the problem at hand, the most recommended [32] approach is to split the data into 70% for training and 30% for test, and this is the ratio we follow in our experiments.

In our problem the split must be carried out based on patient id, not per row or register. The reason for this is that in the dataset there are some patients with more than one X-ray entry, and it will be a clear case of data leaking to have different images belonging to the same patient in different splits.

This patient-based split has two consequences. First, standard cross-validation implementations, which are row-based, could not be used. Thus, we preferred to use a fixed validation set instead of cross-validation. To create this validation set without reducing more the training set, already small due to data limitations, we decided to use half of the patient ids belonging to the test set as validation.

Second, we applied the 70-30 ratio to the number of rows, the ratio in terms of patient ids used for train and validation/test is different, as not all patients have the same number of X-ray images in the dataset.

Taking all this into consideration, our original dataset is split for training, validation, and test purposes as follows:

1. Train: 65% of patient ids.
2. Validation: 17.5% of patient ids.
3. Test: 17.5% of patient ids.

In addition, the split also considers the class of each case, thus preserving the class imbalance ratio over the three sets of data.

Data augmentation techniques are applied to train and validation sets as explained in the next section.

C. Data Augmentation

We have already seen that one of the methods to deal with the problem of a small dimensionality in our available dataset is to use transfer learning to reutilize knowledge extracted from other data, as described in Section IV.A

Other popular tool to reduce the impact of this issue is called data augmentation [33]. As having a large dataset is crucial for the performance of the deep learning model, these tools aim to create synthetic examples based on the original dataset.

There are two main approaches to generate these new artificial samples:

- Generate modifications over the original dataset. The changes applied can be of different nature: affine transformations like rotation and translation, perspective transformations, contrast changes, gaussian noise, dropout of regions, hue/saturation changes, cropping/padding, blurring, etc.
- Create images from scratch based on the global distribution found in the original dataset. For this purpose, Generative Adversarial Networks (GANs) [34] are the state-of-the-art.

We decided to apply rotation and contrast modifications for this experiment to create new images, as they are one of the most common changes you can find among real X-ray images carried out in hospitals.

Therefore, if we decide that the batch size used in each epoch when training the CNN model is for instance 32 images, in each epoch of the CNN training process each one of these 32 images would be the result of randomly selecting one of the original training images and then apply random rotation and contrast modifications to it. Thus, we could say that the data pool when using data augmentation consists of an infinite set of images, all of them variations from the original train data pool images.

D. Hyperparameter Optimization

Each family of Machine Learning models has a set of hyperparameters that are to be optimized to find the optimal model of that family for a given ML task.

Usually this is done by performing a grid search, where you train a different model for each possible combination of hyperparameters you want to analyze, each model is evaluated using a chosen metric over

[2] https://github.com/ieee8023/covid-chestxray-dataset

[3] https://covid19.espacio-seram.com/index.php

a validation set, and the selected hyperparameter values are the ones that correspond to the best performing model. We followed this grid search approach in our experiments.

The detailed list of all the hyperparameters we optimized in our grid search can be found in Table II.

TABLE II. HYPERPARAMETERS OPTIMIZED FOR CHEXNET MODEL USED IN THE FIRST LAYER AND EACH ML FAMILY TRIED AS MODEL IN THE SECOND LAYER

Model	Hyperparameter
CheXNet	epochs
	batch size
	learning rate
Random Forest	number of trees
	nº of candidates at each split
	minimum size of terminal nodes
SVM	cost
	gamma
XGBoost	eta
	gamma
	max_depth
	min_child_weight
	subsample
	colsample_bytree
	num_parallel_tree
	nrounds
	lambda
	alpha
ANN	number of units
	epochs
	batch size
	learning rate

E. Evaluation Metric

As evaluation metric we use the Area Under the Curve (AUC), the most standard evaluation metric for binary classification problems. It is defined as the area under the receiver operating characteristic (ROC) curve, defined by the False Positive Rate (FPR) in the x-axis and the True Positive Rate (TPR) in the y-axis, where:

$$TPR = \frac{TP}{TP+FN} \quad (8)$$

$$FPR = \frac{TN}{TN+FP} \quad (9)$$

where TP, TN, FP, and FN are the true positives, true negatives, false positives, and false negative values, respectively.

F. Experiment Results

AUC results achieved, for both first and second layer models, are presented in Table III. For the case of the COVID-19-CheXNet model, we also show the difference in performance with or without the use of the data augmentation techniques described in Section V.C.

TABLE III. AUC RESULTS FOR EACH MODEL AND DATASET

Model	AUC Train	AUC Val	AUC Test
COVID-19-CheXNet w/o data augmentation	0.93	0.87	0.85
COVID-19-CheXNet w data augmentation	0.93	0.93	0.94
Second Layer	0.99	1	1

Furthermore, the AUC curve obtained by our COVID-19-CheXNet over the test set is shown in Fig. 3.

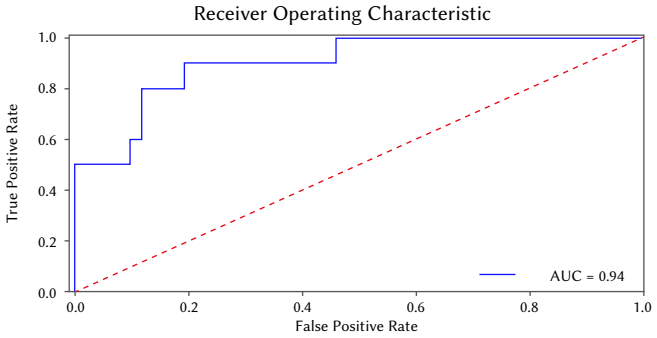

Fig. 3. COVID-19-CheXNet AUC Test. Blue curve represents test AUC for our first layer CNN model predictions. Red dashed line represents a model with an AUC of 0.5 and is used as reference.

We also used heatmaps to visualize which lung areas produced a higher activation in our COVID-19-CheXNet model for deceased patients, which could be useful for practitioner's analysis. One example is shown in Fig. 4.

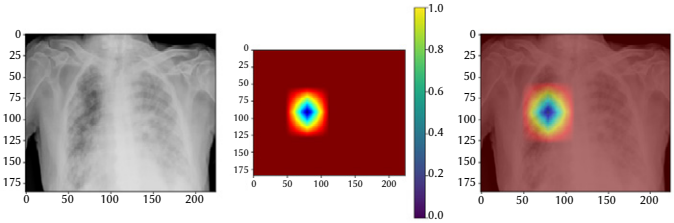

Fig. 4. COVID-19-CheXNet heatmap for a deceased patient.

Finally, we also analyzed the variable importance of each one of the six input variables of the second layer model, by means of conducting a ROC curve analysis on each predictor. Results can be found in Table IV.

TABLE IV. SECOND LAYER MODEL VARIABLES IMPORTANCE IN TERMS OF AUC

Variable	Rank	AUC
pred	1	0.94
age	2	0.71
sex	3	0.63
offset	4	0.57
view	5	0.56
location	6	0.53

VI. DISCUSSION

Several conclusions can be drawn from our experiment results shown in Section V.F. First, all our models achieve a high AUC value, above 0.93 for train and 0.85 over test, which seems to point to a good effectiveness of our transfer learning approach, described in Section IV.A.

In addition, the positive impact of using data augmentation is clear comparing the results of COVID-19-CheXNet with and without applying these techniques. 11% and 7% improvement of AUC is achieved over the validation and test sets, respectively. This shows how data augmentation helps the model to generalize better and not suffer from overfitting problems.

Third, our second layer model can achieve close to perfect performance over the test set. Although the exact AUC values obtained could be impacted by the use of a small dataset and the results should be corroborated once larger volumes of COVID-19 X-ray and outcome data are available, the improvement observed between our

first and second layer models performance shows that our intuition that combining mortality risk prediction based solely on X-ray images with other basic demographic and image information could yield even better predictions seems to be valid.

Finally, variable importance analysis shows that the prediction output of the COVID-19-CheXNet first layer model is clearly the factor with greater prediction power among the six predictors used by our second layer model. The top three is completed with age and sex variables, which seems in line with recent research [35] that have already pointed out them as relevant factors in COVID-19 mortality.

The main contributions of this paper are four. First, we aim to predict the mortality risk of COVID-19 patients based on X-ray images to help clinicians lessen the impact of this disease. Some research has been done on the use of Deep Learning models to diagnose COVID-19 based on this type of images, as reviewed in Section II, but we consider that our model predictions can have a bigger positive impact, as diagnosis can always be done using clinical tests once the patients is in the hospital, as would be the case for a patient suitable of getting an X-ray scan.

Second, we propose to add a second layer to this first model using X-ray images, which will use a combination of the prediction of the first layer DL model and basic demographics of the patients and characteristics of the image. This will allow to further optimize final mortality risk predictions, but it is an approach that has received little attention and no approaches like this are found in the literature about COVID-19 prediction models.

Third, we combined two different sources of data to create a unique and novel COVID-19 dataset, providing X-ray images as well as basic demographic information for a total of 221 registers. Data related to COVID-19 is still rare, so we hope this could help further research.

Finally, we make our model implementations and datasets used in our experiment publicly accessible via GitHub, as detailed in Section IV. Principles of Reproducible Research are always recommended but not always followed, and we wanted to be definitive on this aspect.

VII. Conclusion and Future Work

A. Conclusions

This paper presents a proposed method to predict mortality risk on COVID-19 patients combining a CNN model based only in X-ray images, with a second layer ML model which uses as input the output of that CNN first layer model together with other basic patient demographic and image technical properties information.

Results show that our proposed method achieves close to or even perfect performance regarding AUC over the test dataset used in our experiments.

Furthermore, results also evidence that our proposed techniques, like transfer learning, data augmentation and the addition of a second layer model improve the overall prediction power of the final model, which seems to confirm out hypothesis and the usefulness of our proposed framework.

B. Future Work

We know that the main limitation of our research is the small dataset we were obliged to work with due to COVID-19 data availability. Therefore, conclusions drawn from our experiment results should be confirmed with a different and larger dataset. We are currently collaborating with Hm group of hospital in Spain to use a dataset of more than 2310 patients which we hope could greatly enhance our model power and statistical significance of our conclusions. We hope to have experiment results over this new dataset in the coming months.

Furthermore, a more exhaustive optimization of our models in terms of more layer weights being fine-tuned, additional data augmentation techniques being applied, and a bigger hyperparameter grid search being carried out, can be tested to search for a model performance improvement, and we plan to conduct these experiments with the larger dataset earlier mentioned.

Recent proposed frameworks that allow to mix images input with numeric information in a single CNN are suited to the problem we try to tackle. Experiments using these models could be carried out and results compared with our two-layer proposed framework.

Finally, using GANs as data augmentation tool has been shown to improve results obtained by models in healthcare classification tasks [36], and we aim to test it in our proposed framework.

Acknowledgment

We would like to thank SERAM and the University of Montreal for the public COVID-19 dataset they made available.

References

[1] C. M. Bishop, *Pattern recognition and machine learning*, New York, USA: Springer, 2006.

[2] T. Kimoto, K. Asakawa, M. Yoda, and M. Takeoka, "Stock market prediction system with modular neural networks", in *International joint conference on neural networks*, San Diego, USA, 1990, pp. 1-6.

[3] S. Maes, K. Tuyls, B. Vanschoenwinkel, and B. Manderick, "Credit card fraud detection using Bayesian and neural networks", in *Proceedings of the 1st international naiso congress on neuro fuzzy technologies*, Havana, Cuba, 2002, pp. 261-270.

[4] Y. Gala, A. Fernandez, J. Diaz, and. J. R. Dorronsoro, "Support vector forecasting of solar radiation values", in *Hybrid Artificial Intelligent Systems*, Salamanca, Spain, 2013, pp. 51-60.

[5] J. Prada, and J. Dorronsoro, "General noise support vector regression with non-constant uncertainty intervals for solar radiation prediction", *Journal of Modern Power Systems and Clean Energy*, vol. 6, no. 2, pp. 268-280, 2018, doi: 10.1007/s40565-018-0397-1.

[6] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*, Cambridge, USA: MIT press, 2016.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks", in *Advances in neural information processing systems*, Nevada, USA., 2012, pp. 1097-1105.

[8] H. Kaiming, Z. Xiangyu, R. Shaoqing, and S. Jian, "Deep residual learning for image recognition", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Las Vegas, USA., 2016, pp. 770-778.

[9] R Caruana, and A. Niculescu-Mizil, "An empirical comparison of supervised learning algorithms", *Proceedings of the 23rd international conference on Machine learning*, New York, USA., 2006, pp. 161-168.

[10] R. Gargeya, and T. Leng, "Automated identification of diabetic retinopathy using deep learning", *Ophthalmology*, vol. 124, no. 7, pp. 962-969, 2017, doi: 10.1016/j.ophtha.2017.02.008.

[11] K. Sirinukunwattana, S. Raza, Y. Tsang, D. Snead, I. Cree, and N. Rajpoot, "Locality sensitive deep learning for detection and classification of nuclei in routine colon cancer histology images", *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1196-1206, 2016, doi: 10.1016/ 10.1109/ TMI.2016.2525803.

[12] A. Akselrod-Ballin, L. Karlinsky, S. Alpert, S. Hasoul, R. Ben-Ari and E. Barkan, "A region based convolutional network for tumor detection and classification in breast mammography", Deep learning and data labeling for medical applications, Athens, Greece., 2016, pp. 197-205.

[13] S. Haykin, *Neural networks: a comprehensive foundation*, New Jersey, USA: Prentice Hall PTR, 1994.

[14] C. Chang, and C. Lin, "LIBSVM: A library for support vector machines", ACM transactions on intelligent systems and technology (TIST), vol. 2, no. 3, pp.1-27, 2011.

[15] T. Chen, and. C. Guestrin, "Xgboost: A scalable tree boosting system", in *Proceedings of the 22nd acm sigkdd international conference on knowledge*

discovery and data mining, New York, USA, 2015, pp. 785-794.

[16] L. Breiman, "Random forests", *Machine learning*, vol. 45, no. 1, pp. 5-32, 2001.

[17] S. J. Pan, and Q. Yang, "A survey on transfer learning", *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345-1359, 2009, doi: 10.1109/TKDE.2009.191.

[18] J. Zech, M. Badgeley, M. Liu, A. Costa, J. Titano, and E. K. Oermann, "Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: a cross-sectional study", *PLoS medicine,* vol. 15, no. 11, pp. 962-969, 2018, doi: 10.1371/journal.pmed.1002683.

[19] I. Apostolopoulos, D. Ioannis, and T. A. Mpesiana, "Covid-19: automatic detection from x-ray images utilizing transfer learning with convolutional neural networks", *Physical and Engineering Sciences in Medicine*, vol. 1, no. 1, pp.14-26, 2020.

[20] M. Ahsan, T. Alam, T.Theodore and P. Huebner, "Deep MLP-CNN model using mixed-data to distinguish between COVID-19 and Non-COVID-19 patients", *Symmetry*, vol. 12, no. 9, pp.1526, 2020.

[21] S. Ahuja, B.K. Panigrahi, N. Dey, V. Rajinikanth, and T.K. Gandhi, "Deep transfer learning-based automated detection of COVID-19 from lung CT scan slices", TechRxiv, 2020.

[22] S. Fong, N. Dey, and J. Chaki, "AI-enabled technologies that fight the coronavirus outbreak", *Artificial Intelligence for Coronavirus Outbreak*, 2020, pp.23-45.

[23] N. Cristianini, J. Shaew-Taylor, *An introduction to support vector machines and other kernel-based learning methods*, Cambridge, England: Cambridge university press, 2000.

[24] R. Fletcher, *Practical methods of optimization*, New Jersey, USA: John Wiley & Sons, 2013.

[25] H. Q. Minh, P. Niyogi, and. Y. Yao, "Mercer's theorem, feature maps, and smoothing", in *International Conference on Computational Learning Theory, Pittsburgh*, USA, 2006, pp. 154-168.

[26] N. B. Karayiannis, "Reformulated radial basis neural networks trained by gradient descent", *IEEE transactions on neural networks*, vol. 10, no. 3, pp. 657-671, 1999, doi: 10.1109/72.761725.

[27] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations", in *Proceedings of the 26th annual international conference on machine learning*, New York, USA, 2009, pp. 609-616.

[28] D. Díaz-Vico, J. Prada, and J. R. Dorronsoro, "Deep Support Vector Classification and Regression", in *International Work-Conference on the Interplay Between Natural and Artificial Computation*, Almería, Spain, 2019, pp. 33-43.

[29] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, et al., "Tensorflow: A system for large-scale machine learning", in *12th Symposium on Operating Systems Design and Implementation*, Savannah, USA, 2016, pp. 265-283.

[30] A. Gulli, and S. Palm, *Deep learning with Keras*, Birmingham, UK: Packt Publishing Ltd, 2017.

[31] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, et al., "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning", *Computer Vision and Pattern Recognition*, preprint.

[32] M. Stone, "Cross-validatory choice and assessment of statistical predictions", *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 36, no. 2, pp.111-133, 1974.

[33] J. Wang, and L. Perez, "The effectiveness of data augmentation in image classification using deep learning", *Convolutional Neural Networks Vis. Recognit*, 2017, preprint.

[34] X. Mao, Q. Li, H. Xie, R. Lau, Z. Wang, and. S. S. Paul, "Least squares generative adversarial networks", in *Proceedings of the IEEE International Conference on Computer Vision*, Venice, Italy, 2017, pp. 2794-2802.

[35] J. B. Dowd, L. Andriano, D. M. Brazel, V. Rotondi, P. Block, X. Ding, et al., "Demographic science aids in understanding the spread and fatality rates of COVID-19", *Proceedings of the National Academy of Sciences*, vol. 117, no. 18, pp. 9696-9698, 2020, doi: 10.1073/pnas.2004911117.

[36] E. Wu, K. Wu, D. Cox, and. W. Lotter, "Conditional infilling GANs for data augmentation in mammogram classification", in *Image Analysis for Moving Organ, Breast, and Thoracic Images*, Granada, Spain, 2018, pp. 98-106.

Jesús Prada Alonso

Double Degree in Computer Science and Mathematics at Universidad Autónoma de Madrid, Spain, 2013, double master's in Computational Intelligence and Applied Mathematics, 2015, PhD. in Machine Learning at Universidad Autónoma de Madrid, Spain, 2020. Researcher at Machine Learning Group at Universidad Autónoma de Madrid. Currently working as Machine Learning Manager at Sistemas de Gestión Sanitaria, SIGESA, a Spanish healthcare company, as Machine Learning Specialist at Iberia Express, Spanish airline company, and as professor at Instituto de Empresa, Madrid, and at Escuela de Empresarios, Valencia. Research interests are renewable energy prediction, e-learning, and health, all of them as tasks to solve using Machine Learning or Deep Learning techniques.

Yvonne Gala García

Degree in Mathematics at Universidad Autónoma de Madrid, Spain, 2009, master's in Education, Universidad Autónoma de Madrid, Spain, 2010, master's in Computational Intelligence, Universidad Autónoma de Madrid, Spain, 2012, PhD. in Machine Learning at Universidad Autónoma de Madrid, Spain, currently in progress. Researcher at Machine Learning Group at Universidad Autónoma de Madrid, Spain. Currently working as Machine Learning Manager at Iberia Express, Madrid, and as professor at Escuela de Empresarios, Valencia. Research interests are Machine Learning applied to solar energy prediction, airlines, and health.

Ana Luisa Sierra Bañón

Degree in Biochemistry at Universidad de Navarra, Spain, 2017, master's in Computational Biology, Universidad Politécnica de Madrid, Spain, 2019, master's in Biostatistics, Universidad Complutense de Madrid, Spain, 2019. Currently working as data scientist at Sistemas de Gestión Sanitaria, SIGESA, a Spanish healthcare company. Research interests are computational biology, biostatistics, and Machine Learning applied to health problems.