IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 20 February 2023, accepted 7 March 2023, date of publication 10 March 2023, date of current version 21 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3255834

RESEARCH ARTICLE

Removing Zero Variance Units of Deep Models for COVID-19 Detection

JESÚS GARCÍA-RAMÍREZ[1], **BORIS ESCALANTE-RAMÍREZ**[1,2], **AND JIMENA OLVERES MONTIEL**[1,2]

[1]Facultad de Ingeniería, Universidad Nacional Autónoma de México, Coyoacán, Mexico City 04510, Mexico
[2]Centro de Estudios en Computación Avanzada, Universidad Nacional Autónoma de México, Mexico City 04510, Mexico

Corresponding author: Jesús García-Ramírez (jesus-garcia@cecav.unam.mx)

This work was supported in part by the Universidad Nacional Autónoma de México by means of the Programa de Apoyo a Proyectos de Investigación e Innovación Tecnológica (PAPIIT) under Grant TA101121 and Grant IV100420.

ABSTRACT Deep Learning has been used for several applications including the analysis of medical images. Some transfer learning works show that an improvement in performance is obtained if a pre-trained model on ImageNet is transferred to a new task. Taking into account this, we propose a method that uses a pre-trained model on ImageNet to fine-tune it for Covid-19 detection. After the fine-tuning process, the units that produce a variance equal to zero are removed from the model. Finally, we test the features of the penultimate layer in different classifiers removing those that are less important according to the f-test. The results produce models with fewer units than the transferred model. Also, we study the attention of the neural network for classification. Noise and metadata printed in medical images can bias the performance of the neural network and it obtains poor performance when the model is tested on new data. We study the bias of medical images when raw and masked images are used for training deep models using a transfer learning strategy. Additionally, we test the performance on novel data in both models: raw and masked data.

INDEX TERMS Covid-19 detection, model compression, transfer learning, deep learning explainability.

I. INTRODUCTION

COVID-19 is the disease caused by a new coronavirus called SARS-CoV-2. WHO first learned of this new virus on 31 December 2019, following a report of a cluster of cases of 'viral pneumonia' in Wuhan, People's Republic of China.[1] Different artificial intelligence techniques for Covid-19 detection have been proposed, Deep Learning (DL) techniques are very popular because of their advantages such as the ability to extract features automatically through gradient descent updates [1]. Nevertheless, long training times and models with millions of hyperparameters make these techniques in many cases unfeasible to use in computers with limited resources. Transfer learning (TL) reuses previously obtained knowledge in a similar task and is used to alleviate these limitations [2].

DL is a set of machine learning techniques that uses raw data as input and passes it through multiple abstraction levels, obtaining different representations that are used for classification, regression, and unsupervised, among other tasks [3]. Nonetheless, the explainability of deep models is an important limitation because they are black boxes that obtain different levels of representation and are not interpretable for humans. Some advances of interpretability are reported in literature [4], popular techniques such as GradCam [5] search for regions that a neural network uses for the classification.

Another limitation of DL is that deep models contain a large number of hyperparameters. Because of this, a large number of examples are needed to obtain an acceptable performance in the task. In those cases, TL is an alternative to training a DL agent. In transfer learning previously obtained knowledge in a source task (or more than one source), where the training data is enough to obtain a model with good performance, is used to train a new related task (or to accelerate the training) where the training data is insufficient to train a model [2].

Model compression is used to reduce the hyperparameters and complexity of a trained model, this approach can be divided into four groups: parameter pruning, where

The associate editor coordinating the review of this manuscript and approving it for publication was Massimo Cafaro.

[1] https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers -hub/q-a-detail/coronavirus-disease-covid-19

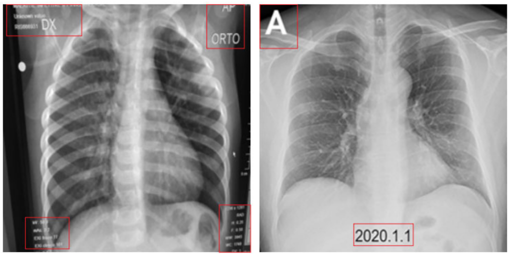


FIGURE 1. Examples of artifacts and metadata printed on the images (shown in red squares). These artifacts could bias the classification of the different available classes. The images are taken from the last version of the Kaggle COVID-19 dataset [9].

the unnecessary units in the model are removed; transferred/compact convolutional filters that try to approximate the output of convolutional layer applying different transformations; low-rank factorization joins different units or layers of a network; and knowledge distillation train less complex models (student model) to mimic the behavior of a teacher model [6].

In this work, we propose a transfer learning strategy using a pre-trained model on ImageNet (in this case, VGG16 [7], but other models can be used as well) in a new task. In this case, Covid-19 datasets are used to evaluate the proposed model. Then, we compress the transferred model removing the units with $\sigma^2 = 0$, that do not contribute to the training of the output Convolutional Neural Network (CNN).

The proposed method learns a new task through fine-tuning, then the obtained model is compressed analyzing the variance of the feature maps and outputs of the fully connected layers. We select a subset of features in the penultimate layer with a greedy search, ranking their importance according to f-test [8]. Finally, we remove those units of the hidden layers that obtain feature maps with $\sigma^2 = 0$ because they do not contribute to the classification of the target. Additionally, we use an explainability technique (GradCam [5]) to determine if a neural network transferred from a pre-trained model on ImageNet to a COVID-19 dataset use the regions of interest to classify an x-ray image or if the metadata or background information is used for the classification. Examples of metadata and artifacts are shown in Figure 1. Also, we study the bias of the model evaluating the obtained models in novel data obtained from different Mexican hospitals.

According to the results, the proposed method can obtain a compressed model with fewer parameters than the source model. Also, it can obtain a subset of the output units that are used to train a less complex classifier than a neural network, such as support vector machine (SVM). We observe that in most cases the trained models pay attention to metadata shown on the images and regions of interest (such as lungs) are not used to determine the class of an image. When we hide this information, the attention of the network changes to the regions of interest, but the performance of the model degrades with respect to the training of the raw data. We share the code of our implementation is a github repository[2]

The main contributions of this work are described next:

- A method for model compression that removes the units that do not contribute to the inference of the classes.
- An algorithm based on a greedy search to obtain a subset of units that serves as input to a classifier such as an SVM that is less complex than a neural network.
- Also, we show that the metadata printed on the images of the COVID-19 Kaggle dataset is used to determine the class of new images. Also, we test the performance of the obtained models on novel data.

The content of the remaining of this paper is the next: Section II describes the background of the paper; in section IV we describe the proposed method for compression of a deep model; experimental results are introduced on section V; finally, in section VI we show the conclusions and future work.

II. BACKGROUND

In this section, we present relevant background information for this paper. First, we describe deep learning, specifically convolutional neural networks that are used in our experiments. Then, we describe transfer learning, which is a scheme used for training agents. Finally, we describe the explainability of deep models.

A. DEEP LEARNING

Deep Learning is a set of methods that use different levels of abstraction for supervised [7], unsupervised [10], and reinforcement learning [11]. Deep models use artificial neural networks to obtain an output updating their weights through gradient descent according to an error function [12].

One of the first applications that used deep models was Alexnet [13] which used a convolutional neural network to classify millions of images in one thousand of classes. One of the main contributions was that they used two graphical processing units to accelerate the training of the networks. This model obtained the best performance on ImageNet 2012 challenge.

VGG16 is a model used in our experiments [7], this model uses small kernels (size $= 3 \times 3$) in different modules that reduce the size of the inputs, summarizing the feature maps using a maxpooling operation. In the last layers, they use fully connected layers to classify the ImageNet dataset. This model outperforms other ones as Alexnet, VGG16 can be seen graphically in Figure 2.

B. TRANSFER LEARNING

Transfer learning (TL) is a type of machine learning where previously obtained knowledge (weights, data, layers, etc.) is used to improve the performance of a new task. In TL exists one ($\mathcal{S}$) or more source tasks with enough data to train a classifier with good performance and a target task ($\mathcal{T}$) that does not have sufficient data to obtain a classifier with an acceptable performance [2]. We aim at obtaining a

[2] https://github.com/gr-jesus/XAI-Covid-19

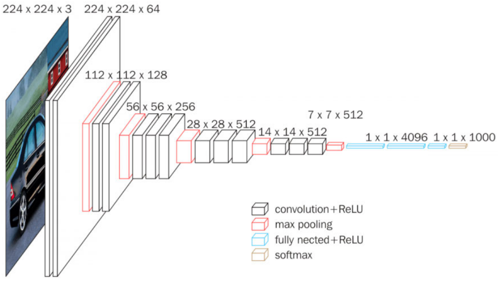


FIGURE 2. **VGG16 model architecture. The model use short kernels (size = 3×3) in different modules to classify ImageNet dataset [7].**

similar performance in less time than training from scratch or obtaining a better performance at the same time.

For transfer learning in neural networks, a fine-tuning scheme is used to obtain a model with better performance than the training from scratch (train an agent initialized with small random weights). Fine-tuning consists of the use of the weights of a pre-trained model on another task to start the training of a new task. Commonly, the last layer is removed and a new one with random weights is added according to the number of classes of the new task.

Also, there exists a transfer learning scheme where the deepest layers of $\mathcal{S}$ in the new task are optimized and another where the full model is updated. The first scheme is used because the deepest layers obtain more specific features of the dataset assuming that it may not be necessary to train every layer on the model [16]. Nevertheless, determining until which layer to transfer or optimize is still an open problem of transfer learning. In this work, we use the second scheme in order not to restrict the performance of the model.

Negative transfer is the main limitation of transfer learning. This appears when training from scratch gets better performance than using a transfer learning scheme. Negative transfer is caused because the distributions (or data) of the source and target data are different. In this paper, we use as the source task a pre-trained model on ImageNet that is a large dataset with a thousand classes in different domains [7], this could help to improve the performance of the target task.

C. EXPLAINABILITY OF DEEP MODELS

Deep learning uses raw images as input to update the weights of a deep neural network through gradient updates using backpropagation algorithm [12]. These updates obtain automatically features from pixels according to the classes in the dataset. Nevertheless, this process obtains abstract features that, in many cases, are not interpretable for humans. It is a different case of other explainable methods such as decision trees or ensemble methods [12], [17].

In the search for interpreting deep models, researchers have proposed different algorithms as [18] and [5] that search for the regions where the network is paying attention to the classification of the images. Zhang et al. [19], propose to build a graph to determine the relevant regions for classification according to the feature maps of a hidden layer. Also, some methods are proposed for natural language processing [20]. In this work, we use GradCam to show the regions used by the deep model to determine the class the image belongs to [5].

III. RELATED WORK

The aim of model compression is to obtain a compressed deep neural network model with the same or similar performance as a model with a higher number of parameters. According to Cheng et al. [6], methods based on model compression can be divided into four groups: knowledge distillation, transferred/compact convolutional filters, low-rank factorization, and parameter pruning and quantization.

Knowledge distillation techniques [21] mimic the outputs of certain layers of a teacher model using a student model with fewer parameters but with the same or similar performance. Most of them try to mimic the logits (outputs of a layer before applying an activation) by applying a relaxing transformation through a softmax function with a temperature value [22], [23]. A previous work [24] proposed a knowledge distillation combined with self-supervised learning formulation to detect COVID-19 disease.

Transferred/compact convolutional filters try to approximate the output of a convolutional layer of a neural network by applying a transformation to obtain a similar representation with another shallower network. These methods can obtain a similar representation but in a new model with fewer parameters [25], [26].

Low-rank factorization tries to obtain low parameters of a deep model joining the units or layers of a network with different transformations of the convolutional kernels in order to obtain a similar representation with a less complex model [27].

Parameter pruning and quantization methods try to reduce the complexity of a deep model by removing the less important units of the model [28]. The proposed method lies in this group, we reduce the number of parameters on a deep neural network removing the units that do not contribute to the classification of a model that is fine-tuned in a new task. Then, we obtain a model with fewer units and the same (and in some cases better) performance as the source model.

Pruning methods to reduce the complexity of neural networks have been proposed in the literature [6], [29], [30]. Choudhary et al. [30] proposed an approach that removes convolutional filters with lower magnitude, hypothesizing that these filters are less relevant than those with higher magnitude. They conducted experiments using computed tomography images, whereas we use X-ray images. In contrast, the authors of [29] proposed an iterative method that prunes a pre-trained model on Imagenet. Unlike this approach, our method prunes a model in one step, and does not require an ensemble method, thus resulting in a simpler inference process for new instances.

IV. COVID-19 DETECTION THROUGH MODEL COMPRESSION

In this section, we describe the proposed method for model compression. This method can be seen graphically on Figure 3, it consists of three stages. In the first stage, a pre-trained model on ImageNet (such as VGG16 [7]) is fine-tuned in the new task, in this work we use Covid-19 datasets. We used the VGG16 model for our experiments because it contains a high number of parameters (132 M) and it is widely used for medical image classification [43], [44]. The second stage consists of a feature selection of the penultimate layer of the fine-tuned model, we experimentally observe that a less complex classifier than a neural network can obtain similar or better performance The last stage consists of removing the units that produce outputs with $\sigma^2 = 0$, those units in most of the cases do not contribute much to the inference of the classes. In the next paragraphs we describe the details of each stage.

In the first stage of the proposed method, we use a pre-trained model on the ImageNet dataset to begin the training of the new task (in this case, datasets for Covid-19 detection). We follow a fine-tuning scheme where the output layer is removed and a new one is added according to the number of classes in the target dataset. Then, the entire model is trained in order not to restrict the performance of the model in the new task.

The second stage of the proposed method consists of a feature selection process using two filters. The feature selection stage is summarized in the Algorithm 1. In the first filter, those features with zero variance ($\sigma^2 = 0$) are removed from the penultimate layer, this means that the features are constant in each instance of the dataset, consequently these features are not useful for classification. In the second filter, we evaluate the number of features using these as input for a new classifier. First, we rank the features using f-test [8]. Then, we evaluate the performance adding some features to the dataset according to the ranking of the f-test. Nevertheless, evaluating each feature could be time-consuming, therefore we use a greedy search adding a hundred features at each step. Finally, we evaluate the range where the higher accuracy is obtained, and we search for the minimum number of features that obtains the best accuracy. Finally, we create a new model with the selected number of features.

Algorithm 1 Method to Compress a Layer of the Pre-Trained Model

```
Require: : A pre-trained on Imagenet $P\_model$, a dataset of the
   target task$D$
Ensure: A compressed model with fewer units in the penultimate
   layer

   Fine-tune $P\_model$ in the $D$
   Extract the output $O$ of a penultimate model in $P\_model$ with $DS$
   Remove those outputs with $\sigma^2 = 0$
   Rank the features using f-test
   while the number of features is high than $O$ do
       Add a hundred features to a new in the $d$
       Evaluate the $d$ in a supervised classifier
   end while
   Select the number of features with the high accuracy
   Search if there is a lower number of features with higher accuracy
   return A compressed model with the selected outputs
```

In the last stage of the proposed method, we observe the variance of the outputs of the hidden layers in the fine-tuned model. In the convolutional layers, we find the variance of the feature maps of each convolutional kernel, and in the fully connected layers the variance of each unit. It is important to take into account that the layers between the flattened and the first fully connected layer have many weights of the model, so it is important to pay attention to the compression of these layers. Also, the deepest layers can be more compressed than the shallow ones, this is because the deepest layers obtain more specific features [16].

After the proposed method is applied we can obtain a compressed model that can be used as a feature extractor on the target task. Then, instead of using a neural network that has a high number of parameters and operations (and is time-consuming in the training stage), a less complex classifier (i.e. Support Vector Machine [31]) can be used in the new task.

The proposed method produces a compressed model that uses fewer parameters compared with the fine-tuned model, preserving similar performance compared with the source model, and in some cases, it obtains better performance. The method is fast because it uses a greedy search in the feature selection stage. Nevertheless, the proposed method is limited in its ability to compress the architecture of the source model, precluding its use in the removal of individual layers or the creation of shallower models, such as those created through knowledge distillation techniques [21]. Additionally, the method is more effective on architectures with fully-connected layers, such as VGG or AlexNet architectures.

V. EXPERIMENTAL RESULTS

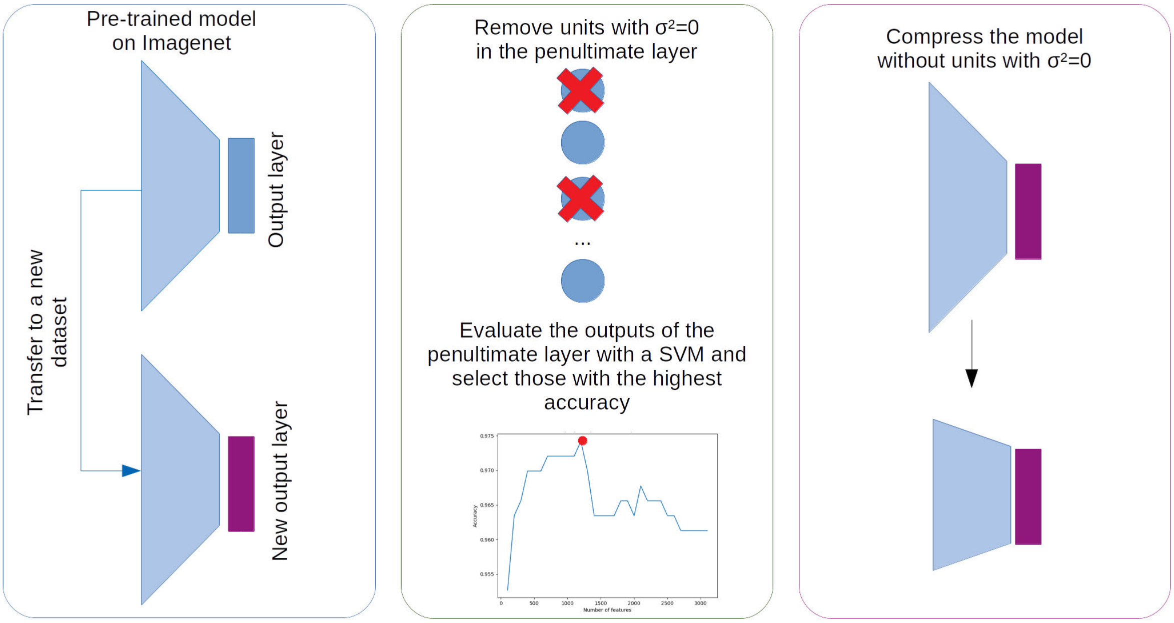
In this section, we present the results of the proposed method and a comparison with other compression model techniques. The aim of the experiments was to show that the proposed method can compress a pre-trained model without degrading the performance of the source model. In this work, we used a VGG16 model pre-trained on ImageNet as the source task. Also, we wanted to show via explanation techniques that the first layers obtained similar features to the source model and in the deepest layers, the source model features change.

For all the experiments we used a computer with Ubuntu 20 operating system, featuring a core i7 processor, 64 GB of memory, and an Nvidia Geforce 3070 for training CNNs. We employed Keras with Tensorflow as the backend for the training of the neural networks and Scikit-Learn [32] for SVMs.

A. DATA DESCRIPTION

In our experiments, we used three datasets for Covid-19 detection hereby described:

- Kaggle v1. This dataset contains 2,905 X-ray images of three classes: covid-19 (219 images), normal (1341 images), and pneumonia (1,345 images). The classes are unbalanced and for covid-19 there are less images than



(1)Fine-tune a pre-trained model on Imagenet

(2) Select the best feature set with SVM

(3) Remove the kernels and outs with σ²=0

FIGURE 3. Proposed method for model compression: (1) fine-tune a pre-trained model on ImageNet dataset, the last convolutional layer is removed and we add a new one with according to the number of classes in the target task; (2) remove those units with $\sigma^2 = 0$ from the penultimate layer, then test the units ranking them with f-test and select the best number of features using a greedy search; (3) remove the units that produce outputs with $\sigma = 0$ from the rest of the pre-trained model.

normal and pneumonia classes. We consider using both versions of this dataset because the new version contains a new class, then the distribution of the datasets changes.

- In the last update of the dataset (kaggle v3) the number of images was increased to 21,165 images of four different classes: covid-19 (3,616 images), normal (10,200 images), pneumonia (1,345 images), and lung opacity (6,062 images). We used the raw data and the masks provided by the authors of [9].[3]
- Also, a dataset with computer tomography images collected from different Mexican hospitals was used for the validation of our method, the dataset contains balanced data for two classes: normal (221 images) and Covid-19 (60 images).

We applied the same pre-processing of the VGG paper [7]. The images were resized to 224 × 224 and the mean RGB value was substracted from each pixel.

B. TRANSFER LEARNING AND MODEL COMPRESSION

The first part of the method consisted of transfering the knowledge of a pre-trained on ImageNet VGG16 model to a new task. In this case, we wanted to learn a model for Covid-19 detection using the datasets previously described. In our experiments, we used Keras library with tensorflow as backend. The datasets were split 20% for testing and 80% for training, and 20% of the training set was used for validation, for the data selection, we follow a stratified strategy for each class in order to use a similar distribution with respect to the entire dataset. We applied a fine-tuning scheme where the output layer was replaced with a new one according to the number of classes on the target datasets. The Adam optimizer was used to train our model with learning rate=0.0001, we used the categorical cross entropy as a loss function and L_2 regularization was used to prevent the overfitting of the model. The model was trained for 20 epochs and we selected the best model according to the accuracy of the validation set.

For testing the second part of the proposed method, we use a SVM [31] using a polynomial kernel with degree $= 3$ and the gamma value is tuned with auto, the rest of the parameters are the proposed on scikit learn [32]. For the greedy search to extract the features in the penultimate layer, we added 100 units at each step. Then, we searched in the previous range where the best accuracy was obtained in order to obtain fewer units. Finally, we removed the units that produced outputs with $\sigma^2 = 0$ and we built a new model without those units. We repeated the experiments ten times in order to report hypothesis tests for the used datasets.

The results of the repetitions of the experiments of the proposed method are reported on Tables 1-4 (Table 1 for kaggle v1, Table 2 for computer tomography dataset, Table 3 for raw data of kaggle v3 dataset and Table 4 for masked data of kaggle v3 dataset). In all the tables we report: the best accuracy of fine-tuning the VGG16 model pre-trained on ImageNet (Fine-tuned); the accuracy of compressing the model with the proposed method and using a SVM to classify the images with a subset of the penultimate layer (Reduced SVM); the features used as input of the SVM (Features);

[3] https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database

TABLE 1. Results of the proposed method in computer tomography dataset.

Fine-tuned	Reduced SVM	Features	Parameters	Percentage	Size (MB)
93.09	92.48	1700	85,322,575	63.54%	332
92.94	93.09	687	97,216,140	72.40%	380
93.71	94.63	1300	107,484,814	80.05%	422
94.32	94.17	808	79,285,218	59.05%	311
93.25	93.86	800	78,680,092	58.60%	306
92.63	92.94	1300	70,654,201	52.62%	274
92.94	92.63	53	40,889,904	30.45%	158
91.56	92.33	837	81,879,564	60.98%	316
89.87	89.11	2100	55,347,373	41.22%	216
91.87	91.87	149	79,132,418	58.93%	306

TABLE 2. Results of the proposed method in kaggle v1 dataset.

Fine-tuned	Reduced SVM	Features	Parameters	Percentage	Size (MB)
97.24	96.90	681	76,729,029	57.14%	301
98.79	98.62	2300	29,847,724	22.23%	117
97.93	97.93	1100	53,135,842	39.57%	208
98.45	98.27	416	53,792,027	40.06%	211
97.24	94.86	249	48,050,922	35.79%	184
97.76	97.24	547	42,263,074	31.48%	166
96.55	95.00	973	70,625,418	52.60%	274
98.45	97.76	1300	52,924,362	39.42%	208
95.35	95.52	2500	28,932,515	21.55%	113
97.41	93.97	1749	37,791,872	28.15%	148

the parameters of the compressed model (Parameters); the percentage (Percentage) of parameters respect the VGG16 model (134,272,835 parameters); and the size of the model in MB (Size).

In our experiments, the best results were obtained with the kaggle v1 dataset. For the best experiment, the model was compressed to 22.15% with respect to the source model and a higher accuracy was obtained (98.62%). Our experiments yielded compressed architectures with 30-66.5 million parameters, which is comparable to the parameter count of architectures such as ResNet101 (44.7 million parameters) [33] and EfficientNet (66.7 million parameters) [34]. However, simpler architectures such as MobileNet (3.5 million parameters) [35] exist. Applying the proposed method, architectures can be compressed to up to 50% of their original parameter count. For the Computer Tomography dataset, the model was less compressed than the kaggle v1 dataset, however, the performance of the compressed model was similar to the source model. The kaggle v3 dataset obtained better results when the raw data was used instead of the masks for the extraction of lung regions, we discuss later the results with these datasets.

It is important to note that our method reduces the size of the model by compressing the source architecture using the training test to prune the less useful units. An ablation study was performed using different numbers of instances to evaluate the obtained model, as shown in Figure 4. It can be observed that the curve increases in a similar way with respect to the number of instances evaluated in the models. The main advantage of our method is the size of the obtained model, as the VGG16 model size is 1.5G, the size of the compressed model with fewer parameters is 113MB.

TABLE 3. Results of the proposed method in Kaggle v3 with raw data.

Fine-tuned	Reduced SVM	Features	Parameters	Percentage	Size (MB)
94.23	94.73	1409	101,555,312	75.63%	396
95.10	95.20	700	80,319,896	59.82%	311
91.49	91.66	2700	58,239,740	43.37%	227
89.48	89.36	1640	84,810,782	63.16%	332
94.16	94.09	334	83,023,758	61.83%	322
94.70	94.75	857	69,776,592	51.97%	274
95.06	94.80	1798	100,072,946	74.53%	393
94.11	93.95	1565	66,559,522	49.57%	261
95.41	95.34	532	88,088,171	65.60%	346
94.25	94.63	3218	105,702,937	78.72%	415

TABLE 4. Results of the proposed method in Kaggle v3 with masked data.

Fine-tuned	Reduced SVM	Features	Parameters	Percentage	Size (MB)
91.56	91.56	1300	45,842,010	34.14%	180
89.55	90.55	793	65,868,677	49.06%	259
91.49	91.66	2700	58,239,740	43.37%	228
89.48	89.36	1640	84,810,782	63.16%	333
89.79	89.77	2322	36,636,762	27.29%	144
90.59	90.40	2329	60,915,460	45.37%	239
91.23	91.44	900	52,637,466	39.20%	206
91.80	92.06	1221	50,501,867	37.61%	198
91.25	91.70	858	65,352,465	48.67%	256
91.44	91.23	2500	43,399,715	32.32%	170

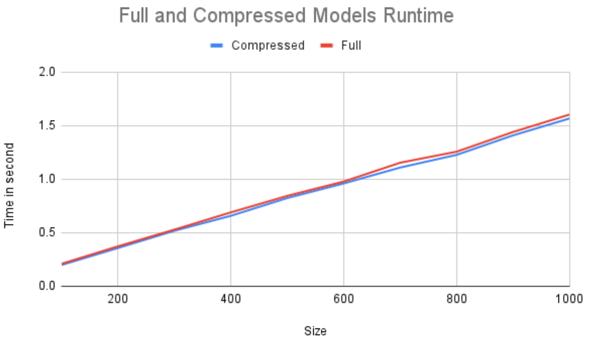


FIGURE 4. Study of evaluating different numbers of instances in the models before and after apply our method. The pruned model is the one with the fewest parameters (28M).

We performed statistical analysis of the results by applying hypothesis tests that are described next. For all the experiments a Kruskal-Wallis hypothesis test was applied [36], we selected this test because it is a non-parametric test, does not assume that the data follows a normal distribution and it requires six elements on each experiment. The results of the hypothesis tests for all the used datasets are shown in Table 5, we set $\alpha = 0.05$ for all tests. It can be seen that there is no statistical evidence that exists a difference between the proposed method and the fine-tuning scheme, the main advantage is that with our method we obtain models with fewer parameters than the source model.

We also compared the proposed method with other methods, using the same reported results of self-supervised knowledge distillation [24] that compared their results with other self-supervised learning methods in Kaggle v3 dataset: BYOL [37], SiamSim [38], Cross [39], PIRL [40] and SimCLR [41].

Table 6 presents the results obtained for the comparison of the proposed method. For sensitivity and specificity, the

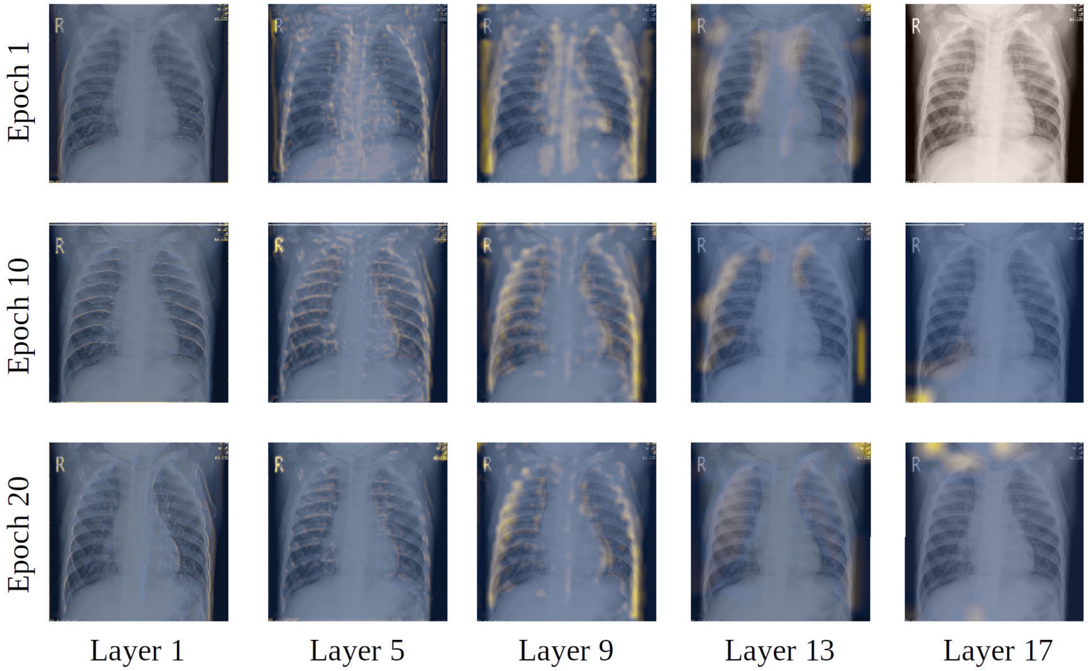


FIGURE 5. Attention of the neural network of different layers during some epochs of the training using GradCam [5]. It can be seen that the attention maps do not change much in the shallow layers. On the other hand, the deepest layers change the attention to other regions that are more relevant for the classification.

TABLE 5. Results of applying the Kruskal-Wallis hypothesis test to the experiments on the four used datasets using $\alpha = 0.05$. In any of the four datasets, there is no statistical evidence of a difference between the trained transfer learning model and the compressed proposed model.

Dataset	p-value
Kaggle v1	0.2252
Computer Tomography	0.9094
Kaggle v3 raw	0.9397
Kaggle v3 masked	0.6772

TABLE 6. Comparison of our method with other approaches.

Method	Sensitivity	Specificity	Accuracy
Ours	0.980	0.995	0.953
Self KD [24]	0.980	0.997	0.957
Cross [39]	0.972	0.997	0.953
BYOL [37]	0.973	0.996	0.953
SimSiam [38]	0.974	0.995	0.950
PIRL-Jigsaw [40]	0.977	0.997	0.951
PIRL-Rotation [40]	0.973	0.997	0.951
SimCLR [41]	0.913	0.994	0.936

Covid-19 class is used as a positive class and the others as negative classes. The proposed method obtained better results than the other methods except with Self-KD. Nevertheless, we obtained a similar performance. it is important to mention that the authors report the need to optimize three models, while our method only requires optimizing one and, moreover, this model is compressed.

We tested the robustness of the obtained models using a novel data set never seen by the model before. We used X-ray images collected from "La Raza" and "Lindavista" hospitals in Mexico City. The dataset contains images corresponding to COVID-19 (60 images) and normal (221 images) classes, so we trained new agents excluding lung opacity and viral pneumonia classes. Then, we tested the performance using this data. We repeated the experiments ten times, the results are shown in Table 7. We can see that the performance of the model trained with masked images was worse than with raw images using the validation set. Nevertheless, when we tested both schemes in novel data, similar scores were obtained. To validate this, a Kruskal-Wallis test with $\alpha = 0.05$ was applied and we obtained p-value $= 0.6230$, consequently, there was no statistical evidence that a difference exists between training with the masked and with raw data. We conclude that any of the two models can be selected and similar performance can be obtained, but when the masked images are chosen we obtain more interpretable features for humans as can be seen in the Figure 6.

TABLE 7. Comparison of the models trained with raw data and masked data. Training with raw data obtains better performance than the training with masked data. Nevertheless, when testing the models with novel data both schemes obtain similar performance.

Raw data		Masked data	
Kaggle V3	Novel data	Kaggle V3	Novel data
98.95	51.24	95.51	74.37
99.13	65.83	97.35	72.95
98.26	54.80	95.18	70.81
98.95	85.40	96.48	63.70
98.87	67.61	95.76	61.56
98.87	74.02	96.63	39.50
98.80	61.20	96.88	86.83
98.80	73.30	95.94	64.76
99.45	66.19	96.56	64.41
99.02	64.41	95.98	88.25

C. EXPLAINABILITY OF THE MODELS

In the proposed method the units in the deepest layers change more that the shallow layers. Shallow layers obtain similar

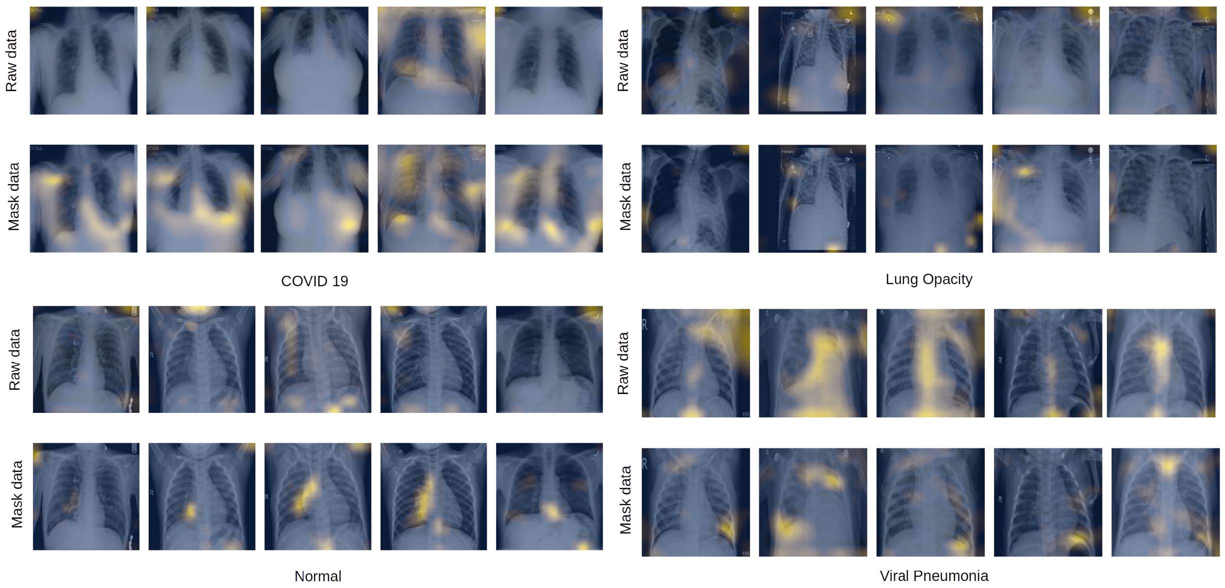


FIGURE 6. Comparison of the two training schemes for the classes in the datasets (Covid-19, lung opacity, normal and viral pneumonia). It can be seen that the model trained with masked images shows more explainable features related to lung regions. On the other hand, raw data pay more attention to metadata and artifacts of the images.

weights compared to the initial model (VGG16 pre-trained on ImageNet). Figure 5 shows the attention of the model during epochs 1, 10, and 20, for different layers in the model. We observe that in the shallow layers the attention of the neural network is similar during the training. On the other hand, the last layer changes to different regions of the image at different epochs of the training.

Also, we analyzed which regions of the images are used to classify the images. We show these regions for both training schemes, using the raw images and using the masked images of Kaggle v3 dataset. The comparison was done using GradCam for the classes in the dataset in Figure 6 (we use deel xplique library to obtain the feature maps [42]). We can see that the network pays attention to different artifacts and metadata printed on the images of Covid-19 and normal class, and in the background for the images that belong to lung opacity. For the viral pneumonia class, the regions are more similar in both datasets.

VI. CONCLUSION AND FUTURE WORK

This work proposed a compression method based on the pruning of units that produce outputs with $\sigma^2 = 0$ after transferring a pre-trained model on ImageNet. Also, we apply a greedy search to select a subset of units in the penultimate layer as a feature selection and use a SVM to classify images of different Covid-19 detection datasets.

The results of the proposed method obtain similar performance to the pre-trained model according to hypothesis tests applied to the used datasets. However, the new model has lower parameters than the source one.

Also, it can be observed that the training stage using the raw data and the masked images, obtains similar performance when we test in novel data. The training using masked images obtain more interpretable features for humans (lung regions), than the training of the raw data that pays attention to the metadata printed on the images.

For future work, we want to explore other methods for model compression like knowledge distillation that can obtain more compressed models than the pruning methods, such as the method presented in this paper. The robustness of the proposed method needs to be evaluated with pre-trained models and architectures, such as VGG19, ResNet and MobileNet. Additionally, we intend to assess the applicability of the proposed method in architectures with diverse components, including residual layers. Moreover, we plan to use pre-trained models with medical datasets to determine if the proposed method yields improved results with similar images.

REFERENCES

[1] Y. Bengio, I. Goodfellow, and A. Courville, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2016.

[2] S. J. Pan, "Transfer learning," *Learning*, vol. 21, pp. 1–2, Jan. 2020.

[3] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[4] A. Singh, S. Sengupta, and V. Lakshminarayanan, "Explainable deep learning models in medical image analysis," *J. Imag.*, vol. 6, no. 6, p. 52, Jun. 2020.

[5] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.

[6] Y. Cheng, D. Wang, P. Zhou, and T. Zhang, "A survey of model compression and acceleration for deep neural networks," 2017, *arXiv:1710.09282*.

[7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[8] S. Lars et al., "Analysis of variance (ANOVA)," *Chemometrics Intell. Lab. Syst.*, vol. 6, no. 4, pp. 259–272, Jul. 1989.

[9] T. Rahman, A. Khandakar, Y. Qiblawey, A. Tahir, S. Kiranyaz, S. B. Abul Kashem, M. T. Islam, S. Al Maadeed, S. M. Zughaier, M. S. Khan, and M. E. H. Chowdhury, "Exploring the effect of image enhancement techniques on COVID-19 detection using chest X-ray images," *Comput. Biol. Med.*, vol. 132, May 2021, Art. no. 104319.

[10] U. Michelucci, "An introduction to autoencoders," 2022, *arXiv:2201.03898*.

[11] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, and J. Veness, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[12] T. M. Mitchell, *Machine Learning*, vol. 1. New York, NY, USA: McGraw-Hill, 1997.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton "ImageNet classification with deep convolutional neural networks," in *Proc. NIPS*. Stateline, NV, USA: Harrahs and Harveys, Lake Tahoe, 2012, pp. 1–12.

[14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. NIPS*. Montréal, QC, Canada: Palais des Congrès de Montréal, 2014, pp. 1–9.

[15] M. E. Taylor and P. Stone, "Transfer learning for reinforcement learning domains: A survey," *J. Mach. Learn. Res.*, vol. 10, no. 7, pp. 1633–1685, 2009.

[16] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 1–9.

[17] A. Cutler, D. R. Cutler, and J. R. Stevens, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2012.

[18] S. Lapuschkin, A. Binder, G. Montavon, K.-R. Müller, and W. Samek, "Analyzing classifiers: Fisher vectors and deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2912–2920.

[19] Q. Zhang, R. Cao, F. Shi, Y. N. Wu, and S. C. Zhu, "Interpreting CNN knowledge via an explanatory graph," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, 2018, pp. 1–10.

[20] K. Qian, M. Danilevsky, Y. Katsis, B. Kawas, E. Oduor, L. Popa, and Y. Li, "XNLP: A living survey for XAI research in natural language processing," in *Proc. 26th Int. Conf. Intell. User Interfaces*, Apr. 2021, pp. 78–80.

[21] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.

[22] C. BuciluG, R. Caruana, and A. Niculescu-Mizil, "Model compression," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2006, pp. 535–541.

[23] J. Ba and R. Caruana, "Do deep nets really need to be deep?" in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 1–9.

[24] G. Li, R. Togo, T. Ogawa, and M. Haseyama, "Self-knowledge distillation based self-supervised learning for covid-19 detection from chest X-ray images," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2022, pp. 1371–1375.

[25] T. Cohen and M. Welling, "Group equivariant convolutional networks," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 2990–2999.

[26] S. Zhai, Y. Cheng, Z. M. Zhang, and W. Lu, "Doubly convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 1–9.

[27] V. Lebedev, Y. Ganin, M. Rakhuba, I. Oseledets, and V. Lempitsky, "Speeding-up convolutional neural networks using fine-tuned CP-decomposition," 2014, *arXiv:1412.6553*.

[28] W. Chen, J. Wilson, S. Tyree, K. Weinberger, and Y. Chen, "Compressing neural networks with the hashing trick," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2285–2294.

[29] S. Rajaraman, J. Siegelman, P. O. Alderson, L. S. Folio, L. R. Folio, and S. K. Antani, "Iteratively pruned deep learning ensembles for COVID-19 detection in chest X-rays," *IEEE Access*, vol. 8, pp. 115041–115050, 2020.

[30] T. Choudhary, S. Gujar, A. Goswami, V. Mishra, and T. Badal, "Deep learning-based important weights-only transfer learning approach for COVID-19 CT-scan classification," *Appl. Intell.*, vol. 53, pp. 7201–7215, Jul. 2022.

[31] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1997.

[32] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Nov. 2011.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[34] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6105–6114.

[35] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.

[36] W. H. Kruskal and W. A. Wallis, "Use of ranks in one-criterion variance analysis," *J. Amer. Stat. Assoc.*, vol. 47, no. 260, pp. 583–621, 1952.

[37] J. B. Grill, F. Strub, F. Altché, and C. Tallec, "Bootstrap your own latent-a new approach to self-supervised learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 21271–21284.

[38] X. Chen and K. He, "Exploring simple Siamese representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 15750–15758.

[39] G. Li, R. Togo, T. Ogawa, and M. Haseyama, "Self-supervised learning for gastritis detection with gastric X-ray images," 2021, *arXiv:2104.02864*.

[40] I. Misra and L. van der Maaten, "Self-supervised learning of pretext-invariant representations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6707–6717.

[41] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.

[42] T. Fel, L. Hervier, D. Vigouroux, A. Poche, J. Plakoo, R. Cadene, M. Chalvidal, J. Colin, T. Boissin, L. Bethune, A. Picard, C. Nicodeme, L. Gardes, G. Flandin, and T. Serre, "Xplique: A deep learning explainability toolbox," 2022, *arXiv:2206.04394*.

[43] M. Mateen, J. Wen, S. Song, and Z. Huang, "Fundus image classification using VGG-19 architecture with PCA and SVD," *Symmetry*, vol. 11, no. 1, p. 1, Dec. 2018.

[44] R. Anand, "Modified VGG deep learning architecture for COVID-19 classification using bio-medical images," in *Proc. IOP Conf., Mater. Sci. Eng.*, 2021, Art. no. 012001.

JESÚS GARCÍA-RAMÍREZ received the degree in computer science engineering from Instituto Tecnológico de Pachuca, the master's degree in computer science from Benemérita Universidad Autónoma de Puebla, and the Ph.D. degree in computer science from Instituto Nacional de Astrofísica, Óptica y Electrónica. He is currently a Postdoctoral Researcher with Facultad de Ingeniería, Universidad Nacional Autónoma de México. His research interests include transfer learning, deep learning, reinforcement learning, and the analysis of medical images.

BORIS ESCALANTE-RAMÍREZ received the Ph.D. degree from the Technical University of Eindhoven, in 1992. He is currently a Professor with Universidad Nacional Autónoma de México. His research interests include bioinspired models for computer vision and image and signal processing.

JIMENA OLVERES MONTIEL received the Ph.D. degree in computer science from Universidad Nacional Autónoma de México (UNAM). She is currently a Professor with UNAM. Her research interests include machine learning, computer vision, and image and signal processing.