

**A low-cost artificial intelligence tool for COVID-19 early medical decision  
making from chest X-ray in underdeveloped countries**

Miguel Mejia<sup>1</sup>, Juan Jaiber Yepes Zapata<sup>1,2</sup>, David Restrepo Rivera<sup>1</sup>, Alejandro Murillo-González<sup>1</sup>, Sebastián Arango<sup>1</sup>, Camilo Pestana<sup>3</sup>, Emmanuel Salinas<sup>4</sup>, J. G. Paniagua<sup>5</sup>, O. L. Quintero<sup>1</sup>, Diana Lucia Serna Higuita<sup>1</sup>, José Julián Garcés Echeverri<sup>1</sup>, Wiston Arrázola<sup>1</sup>, Christian Diaz<sup>1</sup>, and Daniela Marín Ramírez<sup>1</sup>

<sup>1</sup>EAFIT University, Cra 49 No. 7 sur -50, Medellín, Colombia

<sup>2</sup>Universidad de Antioquia, calle 70 No. 52 - 21, Medellín, Colombia

<sup>3</sup>35 Stirling Hwy, Crawley WA 6009, Australia

<sup>4</sup>600 University Avenue, Toronto, Ontario

<sup>5</sup>Instituto Tecnológico Metropolitano, Calle 73 No. 76A-354, Medellín, Colombia

**Author Note**

This work received a grant from Colombian ministry of Science and Technology.

J.G. Paniagua is a Professor in Faculty of Engineering at Instituto Tecnológico Metropolitano, Medellin, Colombia (e-mail: juanpaniagua@itm.edu.co).

Emmanuel Salinas Miranda is a pos-doctoral Fellow at Mount Sinai Hospital Toronto, Canada (e-mail: aemmanuel7@gmail.com).

O. L. Quintero is a Professor in Math Sciences at Universidad EAFIT, Medellin Colombia (e-mail: oquinte1@eafit.edu.co).

All correspondence regarding this paper please send to Miguel Mejía at email:  
mmejiam@eafit.edu.co

## Abstract

This work presents the first approach to the development of a low-cost tool for early decision making by medical personnel using X-ray in underdeveloped countries and the reduction techniques used on the machine learning model in question, in order to improve model footprint in various families of devices without sacrificing accuracy of results. In underdeveloped countries, access to PCR tests for diagnosis requires the use of scientific and technical capabilities in the development of tools that can support medical personnel in this contingency. The Colombian health system and that of several Latin American countries may collapse due to the potential massive assistance to the emergency services. Tools like *Chester* (Cohen et al., 2019) are valuable resources, but lack the availability that is needed in rural and vulnerable areas in third world countries such as Colombia. It was found that compression and/or model reduction techniques were needed (Ajani et al., 2021), as such there was an aim to reduce the architecture used in order to improve model size and memory usage in devices that are available in rural/underdeveloped areas, such as mobile phones of mid-low to mid-high specs. In this approach, we propose a simple but efficient diagnostic application at low-cost for mobile devices following the traditional mobile-development application architecture of MVVM (Model-View-ViewModel). We also present a machine learning model that seeks not only to classify healthy or unhealthy patients, but also to robustly generalize the results for rural care areas where access to high-resolution tomography is impossible. To fulfill the technical requirements of hosting a model in the device, regardless of architecture and specifications, the application of the method of Neural Network Pruning was explored as an effort to reduce model complexity and size both in disk and in memory. This endeavor showed significant results, relying heavily on the *Lottery ticket hypothesis* (Frankle & Carbin, 2018), as well as the works of Paganini and Forde, 2020 as well as Geng and Niu, 2022. This model will be provided to be used in any country, in contrast to works that make use of the application with the model as a private asset. An extension of this tool will be taken to the classification of

healthy patients, patients with bacterial pneumonia and finally patients with pneumonia of viral origin (multi-class learning machine). The model includes a *Model Facts* sheet as suggested by Sendak et al., 2020 for medical apprehension and decision making. In our research this tool is accompanied by another support tool with tomography (via transfer learning). This application is being validated by qualified medical personnel providing feedback for its improvement. Results obtained with the application are promising.

*Keywords:* Medical decision making, Deep Learning, COVID-19, Mobile Application, Diagnosis in vulnerable populations, Medical Diagnosis Tools, Infectious Respiratory Diseases, Deep Learning , Model Reduction , Pruning , Forward Pass Analysis , Model Architecture, Lottery Ticket Theorem

## A low-cost artificial intelligence tool for COVID-19 early medical decision making from chest X-ray in underdeveloped countries

### Abbreviations

- **AI** Artificial Intelligence
- **MVVM** Model - View - ViewModel Architecture
- **GAFA** Google, Amazon, Facebook, and Apple tech companies
- **PA** Posteroanterior position
- **AP** Anteroposterior position
- **HIPAA** Health Insurance Portability and Accountability Act
- **TB** Tuberculosis
- **PNG** Portable Network Graphics
- **RSNA** Radiological Society of North America
- **DL** Deep Learning
- **RGB** Red - Green - Blue
- **DNN** Deep Neural Network(s)
- **CNN** Convolutional Neural Network(s)
- **ReLU** Rectified Linear Unit

### Introduction

The global COVID-19 pandemic had shown that the usual structures for clinical and urgent care were not sufficient when a large scale threat appears. This is true for countries with available accommodations and technological infrastructure, and even more so for

countries where it's not the case for either (e.g. Colombia). The rural and underdeveloped areas of said countries, although less affected in terms of number of cases at the beginning of the pandemic, Bradford et al., 2021 have shown that the number of cases of COVID-19 was risen in rural areas and disproportionately more so than in urban areas. Add to that the lack of medical resources and imaging technology in rural and vulnerable areas, it makes the task of diagnosing and prioritizing patients much more difficult than it already is. In the works by Eshaghi et al., 2021, we find the grave consequences of misdiagnosis in countries with few resources to spare on the medical endeavor. Trying to take advantage of the ubiquity of mobile platforms and the opportunities they provide (Székely et al., 2013), we created an application that will aid the medical staff in the diagnosis of patients, developing a tool that can be used on a mobile device (e.g. smartphone) to search on an Rx image for anomalies, corresponding to the imaging of the patient in question. In accordance with this, we joined efforts with the international community of researchers to help practitioners in a dedicated effort to treat COVID-19 and other pulmonary diseases in an underdeveloped country such as Colombia. We have set out to develop a low cost tool to aid doctors in their diagnosis. The Latin American health systems, including Colombia's is in danger of collapsing due to possible massive visits to the emergency rooms. This work presents an initial development of a low-cost tool using X-ray for early decision making by medical personnel in underdeveloped countries.

## Main goals

This thesis presents the state of the art for classification of X-Ray images on pulmonary diseases and what efforts have been pursued in helping underdeveloped areas of Colombia. It defines an application as a low cost solution to be implemented in areas with low resource allocation which may be in need of an X-Ray analysis tool to detect on a timely manner different pulmonary diseases. It discusses the need to reduce a deep learning model within a mobile framework, as well as the metrics to use when measuring a resulting

model from different techniques aimed at the task. This thesis also discusses in detail the reduction technique of pruning: its mathematical background, algorithms implemented and tested, as well as their respective results and defining further experiments to be implemented and tested within the framework of model pruning (Paganini & Forde, 2020). Finally, an initial state of the developed application is shown, how a medical doctor might use it, and

### Learning Machines in Medical Applications

In the chapter titled *From Artificial Intelligence to Deep Learning in Bio-Medical* in the book *Deep Learners and Deep Learner Descriptors for Medical Applications*, we argued that: *"There is no ambiguity that a machine will ever replace an MD expert, but machine intelligence will benefit and it is aimed at human decision making. Finally, developing solutions is not just training a preconceived convolutional/deep network, neither is machine learning just a toolbox. It requires interdisciplinary work and mathematical background to imagine and formalize a proper feature extractor closest to the human perception of the world."* (Montoya & Paniagua, 2020).

Different papers have been published during the COVID-19 pandemic. Some of them have focused on natural language processing of scientific literature which aims at summarizing the main conclusions to fight the virus. Others have focused on the use of artificial intelligence or machine learning for classification of lung pathology from chest X-ray and computerized tomography (L. Wang & Wong, 2020).

Many scientists have presented their learning machines trained with previous X-ray data-sets and some new samples (compared to what is really needed to build a model that can allow for generalizations) Yet, they do not provide the models to be used on a rapid solution in real life applications. Several works address the architectures and the codes for training along with data-sets. This raises a question: how to develop a tool that can be easily *really* used by a medical doctor? Building a medical assistance tool based on

artificial intelligence is not as easy as training a deep learner with a data-set. Medical doctors from several areas have been developing protocols for assistance based, not only for epidemiological information, but also for patients' co-morbidities and radiology information.

Medical studies and protocols are complex and of the highest possible relevance. Lives depend upon their results. If decisions are just based on a learning machine for diagnosing or defining the treatment of a patient, legal consequences may apply.

According to Sendak et al., 2020 "there has not been a systematic effort to ensure that front-line clinicians actually know how or when to incorporate model output into clinical decisions. Nor is there an expectation that those who develop and promote models are responsible for providing instructions for model use and for the consequences of inappropriate use."

The appearance of a new SARS CoV2 virus that affects human health brings new challenges. We intend to support doctors in making adequate diagnosis and, thus, making more accurate medical decisions regarding the treatment of a patient. Consequently, vIvA-Med Light focuses on generating an application that uses AI and can help us identify suspicious images of a particular pathology generated by the new virus. It can thus serve as a support tool in the diagnosis of the COVID19 disease.

By no means does AI seek to replace the medical community. On the contrary, it intends to be a fundamental support tool for medical decision making using an algorithm that predicts or identifies suspicious lesions in the lung areas. Particularly, it aims at identifying the SARS CoV2 virus producer of the COVID19 disease, using radiological images with chest X-ray in PA position (posteroanterior) or in AP position (anteroposterior) that have already been taken in an upright position in the X-ray room or portable in the room where the patient is located.

As stated in the editorial of Revista Chilena de Radiología (Gálvez, 2017), artificial intelligence will not replace radiologists, but radiologists who use artificial intelligence will replace those who do not. If we generalize, artificial intelligence will not replace doctors,

but doctors who use artificial intelligence will replace those who do not use it. It is not the first time that machines are used to support medical decision making. Back to the origin of radiology in 1895, the electrocardiogram was used to report values and suggest diagnoses that did not replace the clinical eye or diagnosis, but reinforced medical decisions.

## Problem statement

In the search for a quick answer to the needs of Colombian rural areas where no high-level hospitals exist, we aimed to develop a robust, generalizable, not over fitted model to assist X-ray patients' classification. In these hospitals, a radiologist is not always available and medical doctors are not necessarily proficient in X-ray assessment and can be completely overwhelmed by the treatment of COVID-19 patients. Consequently, we gathered a team of radiologists, intensive care medical doctors, medical doctors, mathematicians, and engineers to develop a medical validated model.

The model consists of machine learning that seeks to both classify healthy and unhealthy patients and to robustly generalize the results for rural care areas where access to high-resolution tomography is impossible. An extension of this tool can be used for the classification of healthy patients, patients with bacterial pneumonia, and, finally, patients with pneumonia of viral origin (multi-class learning machine). Our proposal also incorporates a model facts label (Sendak et al., 2020) prepared by the interdisciplinary team who participated in the current research with the purpose of finding a common language between physicians and engineers.

A remarkable contribution to the field was the use of a traditional mobile-development application architecture of MVVM (Model-View-ViewModel) that allowed for a rapid development of the tool. Our main goal was to provide open solutions instead of a brief description that is not useful in reality. Our solution is rigorous because it followed a careful medical review of available databases that can allow for generalizations. We are presently reporting the work in progress and will certainly keep the

scientific community posted of our new developments.

One of the setbacks for this application, as has been discussed by C. J. Wang and Huang, 2013, is the privacy of the patients and the correct management of their information, them being the core tenets in the development of medical applications on mobile platforms. As such, architectural decisions such as the hosting of the model in hardware are taken in order to control and minimize where patient data is exposed and to control the information directly on the hardware, without risking a potential "man-in-the-middle" attack, as described in other reports on the matter, such as the works of Salem et al., 2021 and Melamed, 2018.

### **State of the art of mobile apps for medical prognosis**

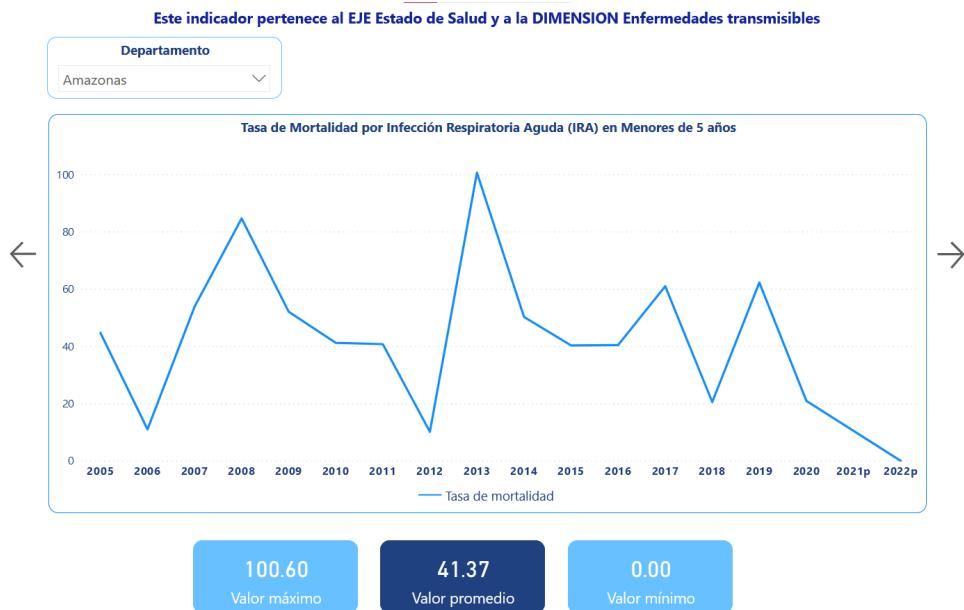
The pandemic has brought about several attempts to aid the medical staff in their prognosis of patients, ranging from online x ray imaging readers (Cohen et al., 2019) ("X-ray app to ease burden on lung specialists and save lives", 2021), applications using also vital signs and laboratory tests results in addition to X ray imaging (Shamout et al., 2021), to efforts on developing a mobile application for multiple diseases ("Mobile chest X-ray analysis uses machine learning models to demonstrate how smart medical apps can be created", 2019). These applications suffer from several issues: Expensive, non-exhaustive, incomplete to our goal for diseases in vulnerable populations: They tend to be expensive, far too large for a limited resource environment (Such as mobile applications) and lack the ubiquity needed for them to be viable in vulnerable areas (Sohn et al., 2020). We need low cost, ubiquitous tools that can help the diagnosis of diseases on people in vulnerable regions (Yadav et al., 2021).

Despite the ubiquity of mobile technologies; such as smartphones, tablets, and such (Székely et al., 2013), mobile applications have not found the appeal for several limitations on the problem at hand: Because of HIIPA laws regarding patient data, it renders the usage of third party cloud hosting applications for the learning machine completely

unusable. One of the reasons being the phrasing of the laws regarding such a topic, as stated by (C. J. Wang & Huang, 2013). For this reason, model inference performed directly on the device is the best option to keep patient privacy. Because of this, the size of the model, number of operations, and performance are limited. In order to accomplish such task, we need to develop an application that balances both inference power and optimization of resources for an environment with such limitations.

### **Current advances regarding COVID-19 and other pulmonary diseases in different populations in Colombia**

According to (Carr, 2020), the surge of COVID-19 patients in health facilities of every kind has lead to a myriad of problems and collateral damage in terms of misdiagnoses of diseases not only of pulmonary affection, but rather multiple kinds (e.g. cardiac arrests), exacerbated by the fact that people were fearful to attend these places, delaying the diagnosis and treatment of the disease in question. This effect is amplified in areas where medical facilities and equipment are scarce, thus leading to an, albeit unintentional, abandonment of vulnerable populations stemming from the remoteness of them (Mueller et al., 2021). This can be seen in reports such as the works by Organization, 2022, where it states: "In the region of the Americas, 1'371.165 cases and 4.158 deaths were reported in EW 25 - a 13.9% increase in cases and a 9.4% increase in deaths compared to the previous week". Looking at the summary collected from 1, 2, and 3 although it has shown an annual decrease in number of reports for deaths of children under 5 years of age due to acute respiratory infections per 100k inhabitants, it's still far higher than we expect it to be: "environmental, constitutional, and social factors may play a critical role in at-home deaths, specifically in those associated with ARI, affecting specific subgroups of families and infants in vulnerable communities"(Caballero et al., 2019).



**Figure 1**

*Timeline of deaths for children under 5 in the Amazonas region caused by acute respiratory infections per 100k inhabitants (en Salud, 2022)*

## Datasets & Learning Machine

### Dataset for COVID-19 X-ray image learning

In order to develop a low-cost application to be used in remote places, we need to be sure the database is approved by medical doctors. As well known, the learning of a mathematical model is not the same as training a mathematical structure. Therefore, many optimization problems can be solved by means of several methods and a set of optimal parameters that fulfill a cost function. Learning a class or multiple classes involves the process of determining the proper sample size and the selection of a mathematical model with a complexity that satisfies the bounds for learning. In pattern recognition in low dimensional spaces, this guarantee is satisfied in most of the cases. However, in pattern recognition in high dimensional spaces, such as images, most models now focus on deep learning machines.

We built a set of databases that included chest X-ray in AP or PA position (not all



**Figure 2**

*Timeline of deaths for children under 5 in the Choco region caused by acute respiratory infections per 100k inhabitants (en Salud, 2022)*

the databases found had images compatible with COVID19). We selected the Radiological Society of North America (RSNA) database from which 19,000 X-ray images were selected. Images with any superimposed artifice or material that was not typical of the body in the lung area (e.g. electrocardiography cables or electrodes, ventilation equipment such as non-rebreathing masks, hoses) was removed. Oxygen, pacemakers, and other superimposed objects in the lung area were also excluded. This was done so that the segmentation of the image that defines the lung, can delimit the cardiac silhouette and bone structures that are removed from the area. Thus, a clean image can be obtained and the pulmonary area can be interpreted as such.

Some of the reviewed databases to use when implementing or transferring learning are described as follows:

1. Chest X-ray images (anterior-posterior) were selected from retrospective cohorts of pediatric patients from one to five years old in Guangzhou Women and Children's

**Figure 3**

*Timeline of deaths for children under 5 in the Guaviare region caused by acute respiratory infections per 100k inhabitants (en Salud, 2022)*

Medical Center, Guangzhou. All chest X-ray imaging was performed as part of patients' routine clinical care. Institutional Review Board (IRB)/Ethics Committee approvals were obtained. The work was conducted complying with the United States Health Insurance Portability and Accountability Act (HIPAA) and was adherent to the tenets of the Declaration of Helsinki (Kermany et al., 2018). An artificial intelligence system using transfer learning techniques was developed. The algorithm was tested in a cohort of pediatric chest radiographs to validate the generalizability of this technique across multiple imaging modalities. The model was then tested with 234 normal images and 390 pneumonia images in jpg format (242 bacterial and 148 viral) from 624 patients. After 100 epochs (iterations through the entire dataset) of the model, the training was stopped due to the absence of further improvement in both loss and accuracy (Kermany et al., 2018). Chest X-ray images (Pneumonia) 5,863 images, 2 categories.

2. Montgomery County (MC) chest x-ray set. The MC set was collected in collaboration with the Department of Health and Human Services, Montgomery County, Maryland, USA. The set contained 138 frontal chest X-ray from Montgomery County's Tuberculosis Screening Program, of which 80 were normal cases and 58 presented manifestations of TB. The X-ray were captured with a Eureka stationary X-ray machine (CR) and were provided in Portable Network Graphics (PNG) format as 12-bit gray level images. The MC set contained manually segmented lung masks for evaluation of automatic lung segmentation methods (Jaeger et al., 2014).
3. RSNA Pneumonia Detection Challenge: The RSNA is an international society of radiologists, medical physicists, and other medical professionals with more than 54,000 members from 146 countries across the globe. They see the potential for ML to automate initial detection (imaging screening) of potential pneumonia cases in order to prioritize and expedite their review. To improve the efficiency and reach of diagnostic services, the RSNA® has reached out to Kaggle's machine learning community and collaborated with the US National Institutes of Health, The Society of Thoracic Radiology, and MD.ai to develop a rich dataset to face this challenge. The chest radiography dataset leveraged to train COVID-Net, to which we will refer as COVIDx, is comprised of 5941 posteroanterior chest radiography images across 2839 patient cases from two open access data repositories.
4. The open source COVIDx data: COVIDx which is comprised of 5941 posteroanterior chest radiography images across 2839 patient cases from two open access data repositories. The open source COVIDx dataset will be leveraged and built upon by researchers and citizen data scientists to accelerate the development of highly accurate, yet practical deep learning solutions for detecting COVID-19 cases, and accelerate the treatment of those who need it the most. Linda Wang, Alexander Wong (Submitted on 22 Mar 2020).

## ChexNET Learning Machine

In order to reply to medical doctors' potential needs, we preferred to perform transfer learning on a known model (Rajpurkar et al., 2017). The main problem was the lack of availability of the original one. Therefore, the entire training process had to focus on teaching the model how to represent at least two classes. Related to COVID-19, the useful pathology to be detected by X-ray was viral-pneumonia. Then, the training set for this task was a real quest. After a medical curation of the data-set (Kermany et al., 2018) (Jaeger et al., 2014) (L. Wang & Wong, 2020) and several attempts to modelling, we developed a useful model to help the improvement of health services, assisting medical doctors with no access to computerized tomography on early decision-making processes based on the architecture of the CheXNET model (Rajpurkar et al., 2017).

CheXNet is a model trained on the ChestX-ray14 dataset (X. Wang et al., 2017), which contains 112,120 frontal-view chest X-ray images individually labeled with up to 14 different thoracic diseases, including pneumonia. For the CheXNet architecture, a DenseNet-121 (Huang et al., 2016) model, pre-trained on ImageNet dataset(Krizhevsky et al., 2012), was used. Transfer learning was used to repurpose it using the new dataset. This model is a 121-layer convolutional neural network trained on ChestX-ray 14, currently the largest publicly available chest X-ray dataset, containing over 100,000 frontal-view X-ray images with 14 diseases (Rajpurkar et al., 2017). CheXNet exceeds average radiologist performance on the F1 metric. CheXNet detects all 14 diseases in ChestX-ray14 and achieves state of the art results on all 14 diseases. CheXNet is a 121-layer Dense Convolutional Network (DenseNet) (Huang et al., 2016) trained on the ChestX-ray 14 dataset. DenseNets improve flow of information and gradients through the network, making the optimization of very deep networks tractable. Researchers replace the final fully connected layer with one that has a single output, after which they apply a sigmoid nonlinearity (Rajpurkar et al., 2017).

Based on the analysis performed by our team of medical doctors, in which not all

the classes of CheXNet were useful or valuable for the real life practice under the COVID-19 actual situation, we proceeded in a different way. We decided to simplify the learning problem to two classes (healthy and not healthy), promoting the likelihood of feasibility validation in real life practice. Consequently, we used the CheXNet architecture because the model was not actually available for the application implementation and retraining to the classes mentioned.

CheXNet architecture uses a DenseNet-121 architecture trained on the Chest-14 dataset. We used a pre-trained (ImageNet) DenseNet-121 network as a baseline and fine-tuned it on the Chest-14 dataset. Similarly, DenseNet-169, DenseNet-201 and ResNet-101 (Xie et al., 2016) were fine-tuned on the same dataset. Their per-class results are compared on table 1. In addition, Class Activation Maps (CAM) (Zhou et al., 2015) were used to evaluate the qualitative results.

### Changes to ChexNET Learning Machine

One of the setbacks for this application, as has been discussed by C. J. Wang and Huang, 2013, is the privacy of the patients and the correct management of their information, this being one of the core tenets in the development of medical applications on mobile platforms. As such, architectural decisions such as the hosting of the model in hardware are taken in order to control and minimize where patient data are exposed and to control the information directly on the hardware, without risking a potential "man-in-the-middle" attack; as described in other reports on the matter such as (Salem et al., 2021) and (Melamed, 2018).

Reducing the application footprint entails maintaining a base-level performance of the model in question, while optimizing the usage of resources within the device. For this reason, the resulting model needs to fulfill the following requirements:

1. *Smaller size in disk:* Given the application is to be used on a mobile device, the variability of available storage on the devices forces us to seek a smaller application

**Table 1**

*Comparison performance using ROC curve metric for CheXNet, DenseNet-169, DenseNet-201 and ResNext-101 architecture*

	CHEXNET (Baseline)	DenseNet-169	DenseNet-201	ResNext-101
CLASS	ROC_AUC			
Atelectasis	0.82	0.84	0.84	0.73
Cardiomegaly	0.91	0.91	0.91	0.65
Effusion	0.88	0.89	0.89	0.84
Infiltration	0.71	0.72	0.72	0.66
Mass	0.85	0.86	0.87	0.67
Nodule	0.78	0.81	0.81	0.63
Pneumonia	0.77	0.78	0.78	0.69
Pneumothorax	0.88	0.89	0.89	0.75
Consolidation	0.81	0.82	0.82	0.78
Edema	0.89	0.90	0.90	0.83
Emphysema	0.93	0.94	0.94	0.70
Fibrosis	0.84	0.86	0.86	0.71
Pleural Thickening	0.78	0.80	0.80	0.68
Hernia	0.93	0.95	0.93	0.79

size, thus reducing all assets of it, model included.

2. *Simpler architecture:* Given the high variance in available resources across mobile devices (since there is no way to know for sure what hardware the application will be installed in), in order to reduce the model complexity, we need to find a small, simple architecture, out of which the small size in disk will follow given the model topology, a.k.a. Shape of network and number of connections.
  
3. *Small variance in performance metrics:* The ensuing model needs to achieve equal-or-faster performance than its un-reduced counterpart in the metrics evaluated in question: **accuracy**, **precision**, **recall**, **f1-score**, and **AUROC**.

## Experiments in architecture

In order to reduce the architecture print of the model within the hardware, we first needed to determine a baseline from which all reductions would be calculated. By knowing the baseline of the model we're currently working with, we're able to grasp what the changes to the models entail. The properties of the model unchanged model are as follows:

- **Number of operations:** 3.7637 GMac, or 7.5274 GFlops
- **Number of parameters:** 6'968.206 parameters
- **Size in disk:** Estimated size in disk of 321.35MB

These are the parameters to be changed and to evaluate in order to determine the success of these endeavours.

A number of models and iterations were tested when training our implementation of the models. These implementations differ in level of specialization of the model and number of epochs (i.e. times a whole dataset is iterated over).

We analyzed the performance of the current architecture for several classification methods, separating images in Healthy/Not Healthy classes, and further dividing the Not Healthy classes into the ones we are interested in. Analyzing the mapped features when forwarding an image on the model, we find several instances where the model pays attention to elements outside of the lung area, as well as elements that lack importance in neuron activation for the subsequent layers of the model. The following applications were developed and tested:

1. Changing a Denselayer to a regular convolutional layer.
2. Changing the DenseNet model to an Xception architecture (as discussed by Chollet, 2017).
3. Combining Densenet and ASPP architectures (ASPP based on work by Chen et al., 2016) by removing a final Denseblock and combining it with an ASPP layer.

4. Pruning random convolutional layers from a given Denselayer (as discussed by Paganini and Forde, [2020](#)).
5. Pruning convolutional layers from a pool based on L1-metric to eliminate global lowest (as discussed by Paganini and Forde, [2020](#)).
6. Iterate over pruning algorithm, eliminating a fixed percentage of the elements to be determined (as discussed by Paganini and Forde, [2020](#)).
7. Iterate over pruning algorithm, increasing the percentage of layers to be eliminated from the mobile in question.

### Analyzing information flow on forward and backward passes

In order to determine the impact of the architecture changes performed in the experiments, we need to understand the information flow of the images analyzed by the model, and how it propagates throughout the network in order to determine what the model is paying attention to. To accomplish this, the following techniques were applied:

- *Pixel Activation Count:* Let  $T$  be a  $3d$  array with dimension  $n \times w \times h$ , corresponding to the output of a given part of the neural network, e.g. the result of a convolutional layer with a given input. This array can be seen as a series of  $n$  images with dimensions  $w \times h$ . Each of the  $n$  images is passed through a Rectified Linear Unit (ReLU) activation function, which has the effect of having all values less than or equal to 0 go to 0, while everything else stays the same. The ReLU activation function is defined as follows:

$$\text{ReLU}(x) = \begin{cases} x, & \text{if } x > 0 \\ 0, & \text{if } x \leq 0 \end{cases}$$

Let  $p_i$  be a pixel on the  $i - th$  image with coordinates  $(j, k)$ , where  $1 \leq i \leq n$ ,  $1 \leq j \leq w$ ,  $1 \leq k \leq h$ . Let  $T'$  be a new  $3d$  array resulting from the execution of the

ReLU function on all pixels on all images of  $T$ , that is

$T' = \text{ReLU}(T) = \text{ReLU}(p_i), i = 1, \dots, n$ . Let  $C$  be a  $3d$  array of dimensions  $n \times w \times h$  defined as follows: Let  $p_{C_i}$  be the pixel of the  $i - th$  image on array  $C$  with position  $(j, k)$ , we can see that there is a direct parallel with pixel  $p_i$  of array  $T$ , we define the former as

$$p_{C_i} = \begin{cases} 1, & \text{if } p_i > 0 \\ 0, & \text{otherwise} \end{cases}$$

It can be seen that the value of  $p_{C_i}$  is equal to the derivative of the ReLU function evaluated at  $p_i$ , that is  $C = \frac{d[\text{ReLU}(x)]}{dx}(T)$ . Lastly, let  $S$  be a  $2d$  array of dimensions  $w \times h$ , defined as follows: Let  $p_S$  be the pixel on position  $(j, k)$  of the image defined by  $S$ , we define it as

$$p_S = \sum_{i=1}^n p_{C_i}$$

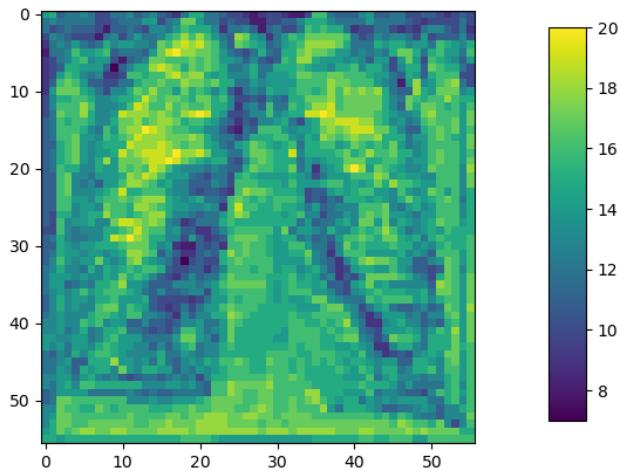
This effectively counts the amount of 1s found on  $C$ , or conversely, counts the amount of times that pixel  $p_i$  on  $T$  was activated via the ReLU function on  $T$ . This creates a  $3d$  "histogram" of the activation zones in question, and how they behave for each specific image. Figures 4 and 6 correspond to such histograms for 2 different blocks within the neural network on an image with bacterial pneumonia, whereas figures 5 and 7 correspond to those same parts of the network on an image with viral pneumonia (possible COVID).

By analyzing the activation count in forward pass on the network, we can determine the relevance of layers within the model, and how the reduction of the algorithm can aid in the passage of relevant information to deeper layers of the model. This method allows us to verify the parts of the image that a specific layer or block is paying attention to, and to mark as candidates those who have less activation counts than other same level parts within that model.

- *Backpropagation Gradient Analysis:* The rationale behind this method comes from the fact that the parameters within the network need to be updated once new information enters the model. These parameters are updated by a certain amount, which is proportional to the difference of the error when evaluating the error function of the model w.r.t. the output of said model.

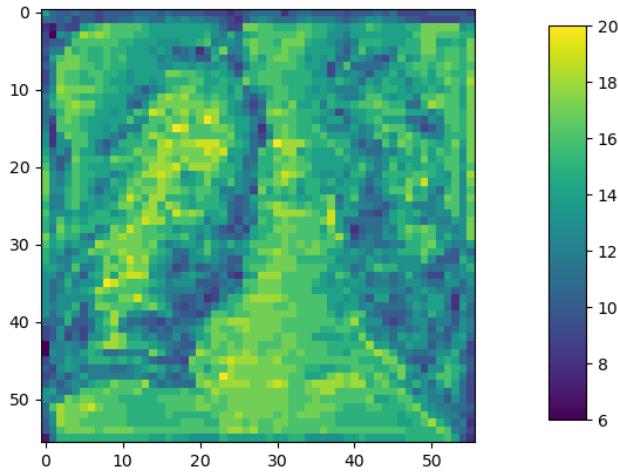
Let  $\nabla\theta_i$  be a vector of size  $p \times q \times r$  as the gradient for the  $i - th$  layer of the model, we can determine the amount of change that layer needs to undergo by taking  $|\nabla\theta_i|$  and adding all  $p$  layers, this will give us a matrix of size  $p \times q$  containing the sum of the total amount of change for all the elements of layer  $i$ . Doing this process for all candidate layers on the model gives us a series of  $p \times q$  matrices, which can be interpreted as images and analyzed visually.

Figures 9 and 8, alongside figures 11 and 10 represent such visualizations for the same layer applied on different kinds of images (e.g., different pathologies).



**Figure 4**

*Image histogram showing the pixels that are activated most times when passing through the first layer of the first denseblock when dealing with an image of bacterial pneumonia*



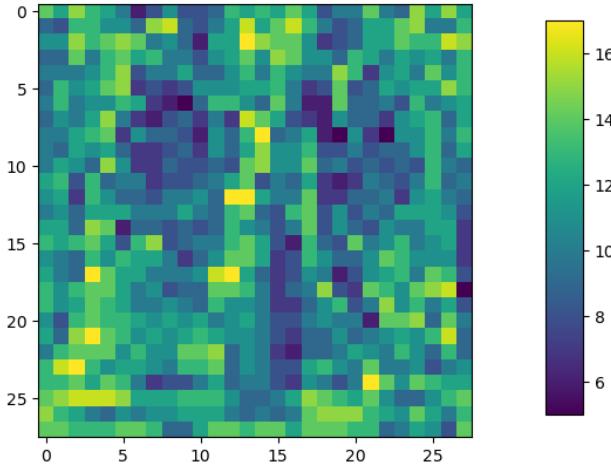
**Figure 5**

*Image histogram showing the pixels that are activated most times when passing through the first layer of the first denseblock when dealing with an image of viral pneumonia*

Understanding how the model interprets the information contained within the images, we have successfully determined the characteristics of the baseline architecture, allowing us to perform an exhaustive comparison between the resulting architectures and giving us the tools to choose one that fulfills the requirements discussed above.

### Model Pruning

As stated in the experiments above, methods involving model pruning were evaluated. In order to understand the results for these, we need to understand first what model pruning entails. As explained by See et al., 2016, model pruning consists on the elimination of neurons or connections from a Deep Learning Neural network, with the goal of creating a smaller and simpler version of the original architecture, with little performance loss, aiming to reduce redundancy within a deep learning model. In the case of Convolutional Neural Networks, as explained by Z. Wang et al., 2021, channels are eliminated instead of "neurons", however the principle still stands. In order to determine the weights or neurons to be removed, we need a function to evaluate them and a removing



**Figure 6**

*Image histogram showing the pixels that are activated most times when passing through the first layer of the second denseblock when dealing with an image of bacterial pneumonia*

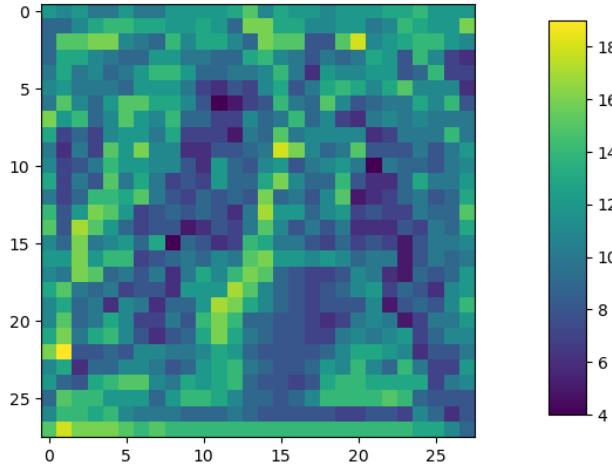
criterion. For this implementation, a selection of  $k\%$  of weights is used, evaluated as those with the lowest value of their  $\mathbf{L}_n$  norm.

### Formal definition and Lottery Ticket Hypothesis

As stated by Frankle and Carbin, 2018, we can find from a family of models  $\mathcal{M}$ , and based on a model  $M$  from that given family, a subnet  $M'$  such that, for any given input  $\mathcal{J}$  and a function of evaluation  $\mathcal{F}$  the following holds true:

$$\forall_{M \in \mathcal{M}} \exists_{M' \in \mathcal{M}} (|M| \geq |M'|) \longrightarrow \forall_{\epsilon > 0} \exists_{\mathcal{F}} (\mathcal{F}(M(\mathcal{J}), M'(\mathcal{J})) \leq \epsilon) \quad (1)$$

From equation 1 we can gather that, whenever we find a model  $M'$  from a given family of models  $\mathcal{M}$ , having less parameters than a base model  $M$ , we can find a function to evaluate the performance of said models on a given input  $\mathcal{J}$  such that the performance is arbitrarily close. That is, the evaluation on both models are arbitrarily close, meaning we can find a smaller model  $M'$  with a similar performance to  $M$ .



**Figure 7**

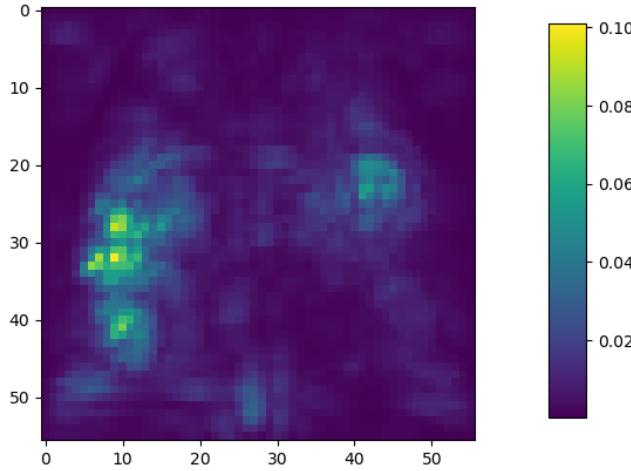
*Image histogram showing the pixels that are activated most times when passing through the first layer of the second denseblock when dealing with an image of viral pneumonia*

We take advantage of this fact from equation 1 to design a pruning algorithm to iterate over the various convolutional layers from the dense layers of blocks 1 through 4 of the network, as can be found on works such as Paganini and Forde, 2020 and Frankle and Carbin, 2018. An iterative re-initialization process will achieve greater results as well as per Paganini and Forde, 2020. We define then 3 pruning algorithms, to create different lottery tickets and compare them in order to determine the best approach in terms of pruning decisions.

### Pruning layer selection algorithms

The algorithms, regardless of how a layer is selected, will take the weights from that specific layer and set to 0 the proportion of those who correspond to the lowest values. The proportion of weights to set to 0, as well as the selection criterion, are hyper-parameters of the algorithm. For our experiments we used the following hyperparameters:

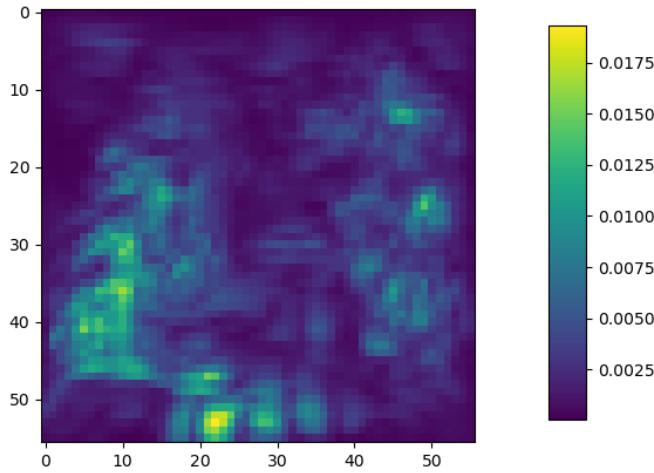
The algorithms designed are as follows:



**Figure 8**

*Image showing the pixels that are activated during the backward pass of the model, showcasing places within the image that trigger the largest change between gradients on the first layer of the first denseblock*

1. *No Pruning:* Model baseline using the hyperparameters listed in 2, using ChexNet model Rajpurkar et al., 2017 and using dataset ChestX-ray14 Rajpurkar et al., 2017, using algorithm 1.
2. *Single Pruning:* Resulting model after performing a single pass of weight elimination using pruning, using hyperparameters listed in table 2, using algorithm 2.
3. *Iterative Pruning:* Resulting model after performing single pruning algorithm and re-train  $n$  number of times,  $n$  being the number of pruning epochs, as defined in table 2, using algorithm 3.
4. *Incremental Pruning:* Resulting model after performing single pruning algorithm and re-train incrementing each time the percentage  $k_{init}\%$  of parameters being pruned by a factor of  $\delta_k$  until a final pruning of percentage  $k_{end}\%$  is achieved.  $k_{init}\%$ ,  $\delta_k$ , and  $k_{end}\%$  are hyperparameters Prune Percentage Start, Prune Percentage Step, and



**Figure 9**

*Image showing the pixels that are activated during the backward pass of the model, showcasing places within the image that trigger the largest change between gradients on the first layer of the first denseblock when dealing with an image of viral pneumonia*

Prunce Percentage End, respectively, all of them defined in table 2, using algorithm 4.

## Experiment methodology and results

While performing the different experiments defined above, mixed results were found when executing the changes to layers on the original architecture, due to one or several metrics failing.

**Changing a Denselayer to a regular convolutional layer.** This approach reduced number of parameters in overall model, reduced model size. Performed poorly on ChexNet database and on transfer learning evaluations. Rejected due to poor performance in both evaluations.

**Changing the DenseNet model to an Xception architecture.** This approach increased number of parameters in overall model, augmented model size. Performed well on ChexNet database and on transfer learning evaluations. Rejected due to

---

**Algorithm 1** Training algorithm

---

**Require:**  $num\_epoch > 0$

**Require:**  $epoch\_stop \leq num\_epoch$

```

function TRAIN( $num\_epoch, epoch\_stop, model$ )
     $current\_epoch \leftarrow 0$ 
     $epochs\_no\_improvement \leftarrow 0$ 
     $best\_accuracy = 0$ 
    while  $current\_epoch < num\_epoch$  and  $epochs\_no\_improvement < epoch\_stop$  do
         $train\_model(model)$ 
         $current\_accuracy \leftarrow evaluate(model)$ 
        if  $current\_accuracy \leq best\_accuracy$  then
             $epochs\_no\_improvement \leftarrow epochs\_no\_improvement + 1$ 
        end if
    end while
    return  $model$ 
end function

```

---



---

**Algorithm 2** Single pruning algorithm

---

**Require:**  $0 < prune\_percent \leq 100$

**Require:**  $prune\_method \in \mathbf{L}_n norms$

```

function SINGLE_PRUNING( $prune\_percent, prune\_method, num\_epoch, epoch\_stop, model$ )
     $trained\_model = train(num\_epoch, epoch\_stop, model)$ 
     $pruned\_model = prune(prune\_percent, prune\_method, trained\_model)$ 
    return  $pruned\_model$ 
end function

```

---

---

**Algorithm 3** Iterative pruning algorithm

---

**Require:**  $0 < \text{prune\_percent} \leq 100$

**Require:**  $\text{prune\_method} \in \mathbf{L}_n \text{norms}$

**Require:**  $\text{prune\_epochs} > 0$

**Require:**  $\text{prune\_early\_stop} \leq \text{prune\_epochs}$

```

function ITERATIVE_PRUNING( $\text{prune\_percent}$ ,  $\text{prune\_method}$ ,  $\text{prune\_epochs}$ ,  $\text{prune\_early\_stop}$ ,  $\text{nu}$ 
 $\text{current\_prune\_epoch} \leftarrow 0$ 
 $\text{pruning\_no\_improvement} \leftarrow 0$ 
 $\text{best\_accuracy} = 0$ 
while  $\text{current\_prune\_epoch} < \text{prune\_epochs}$  and  $\text{pruning\_no\_improvement} <$ 
 $\text{prune\_early\_stop}$  do
     $\text{model} = \text{train}(\text{num\_epoch}, \text{epoch\_stop}, \text{model})$ 
     $\text{model} = \text{prune}(\text{prune\_percent}, \text{prune\_method}, \text{model})$ 
     $\text{current\_accuracy} \leftarrow \text{evaluate}(\text{model})$ 
    if  $\text{current\_accuracy} \leq \text{best\_accuracy}$  then
         $\text{pruning\_no\_improvement} \leftarrow \text{pruning\_no\_improvement} + 1$ 
    end if
     $\text{current\_prune\_epoch} = \text{current\_prune\_epoch} + 1$ 
end while
return  $\text{model}$ 
end function

```

---

---

**Algorithm 4** Incremental pruning algorithm

---

**Require:**  $0 < \text{prune\_percent\_start} \leq 100$

**Require:**  $0 < \text{prune\_percent\_end} \leq 100$

**Require:**  $\text{prune\_percent\_end} > \text{prune\_percent\_start}$

**Require:**  $0 < \text{prune\_percent\_step} \leq \text{prune\_percent\_end} - \text{prune\_percent\_start}$

**Require:**  $\text{prune\_method} \in \mathbf{L}_n\text{norms}$

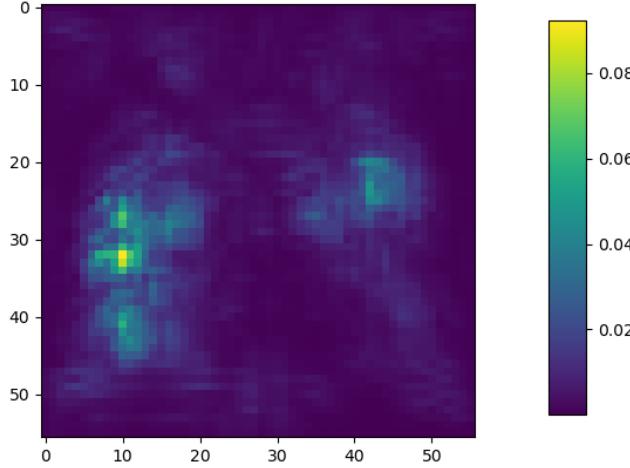
**Require:**  $\text{prune\_early\_stop} \leq \text{prune\_epochs}$

```

function ITERATIVE_PRUNING( $\text{prune\_percent\_start}, \text{prune\_percent\_step}, \text{prune\_percent\_end}, \text{prune\_early\_stop}$ )
     $\text{current\_prune\_percent} \leftarrow \text{prune\_percent\_start}$ 
     $\text{pruning\_no\_improvement} \leftarrow 0$ 
     $\text{best\_accuracy} = 0$ 
    while  $\text{current\_prune\_percent} \leq \text{prune\_percent\_end}$  and
         $\text{pruning\_no\_improvement} < \text{prune\_early\_stop}$  do
             $\text{model} = \text{train}(\text{num\_epoch}, \text{epoch\_stop}, \text{model})$ 
             $\text{model} = \text{prune}(\text{current\_prune\_percent}, \text{prune\_method}, \text{model})$ 
             $\text{current\_accuracy} \leftarrow \text{evaluate}(\text{model})$ 
            if  $\text{current\_accuracy} \leq \text{best\_accuracy}$  then
                 $\text{pruning\_no\_improvement} \leftarrow \text{pruning\_no\_improvement} + 1$ 
            end if
             $\text{current\_prune\_percent} = \text{current\_prune\_percent} + \text{prune\_percent\_step}$ 
        end while
        return  $\text{model}$ 
end function

```

---



**Figure 10**

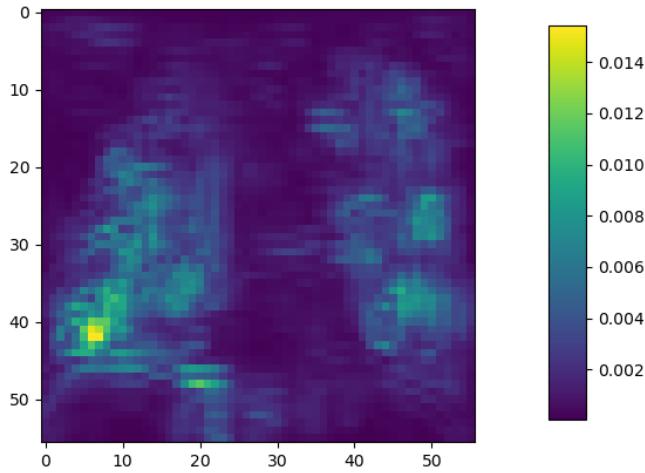
*Image showing the pixels that are activated during the backward pass of the model, showcasing places within the image that trigger the largest change between gradients on the second layer of the first denseblock when dealing with an image of bacterial pneumonia*

increase of overall model, which goes against the goal of this work.

**Combining Densenet and ASPP architectures by removing a final Denseblock and combining it with an ASPP layer.** This approach increased number of parameters in overall model, augmented model size. Performed poorly on ChexNet database and on transfer learning evaluations. Rejected due to poor performance on both metrics evaluated.

However, evaluating the results for pruning algorithms, yielded more promising results in terms of maintaining a baseline accuracy with respect to the original (un-pruned) model

1. Pruning random convolutional layers from a given Denselayer (as discussed by Paganini and Forde, [2020](#)).
2. Pruning convolutional layers from a pool based on L1-metric to eliminate global lowest (as discussed by Paganini and Forde, [2020](#)).



**Figure 11**

*Image showing the pixels that are activated during the backward pass of the model, showcasing places within the image that trigger the largest change between gradients on the second layer of the first denseblock when dealing with an image of viral pneumonia*

3. Iterate over pruning algorithm, eliminating a fixed percentage of the elements to be determined (as discussed by Paganini and Forde, [2020](#)).
4. Iterate over pruning algorithm, increasing the percentage of layers to be eliminated from the mobile in question.

These approaches maintained number of parameters in overall model (Changed weights to 0 do not decrease number of parameters in model, as the 0 still needs to be stored), however due to number of 0s in parameters, the compressed size of the model decreased. These models performed well when evaluated Area Under ROC curve (Fawcett, [2006](#)), however when evaluating them after executing transfer learning, these models showcase the same metrics.

Model training metrics show that both pruned and unpruned methods behave similarly in training evaluations. This is result of the similarity in their algorithms, given that for pruning training of the model, a number of iterations of regular training need to be

**Table 2**

*Hyperparameter values used on the different training algorithms*

<b>Hyperparameter</b>	<b>Value</b>
<b>Batch Size</b>	6
<b>Number of Epochs</b>	10
<b>Epochs for Early Stop</b>	3
<b>Number of Pruning Epochs</b>	3
<b>Epochs for Early Stop when Pruning</b>	2
<b>Prune Percentage Start</b>	30%
<b>Prunce Percentage End</b>	50%
<b>Prunce Percentage Step</b>	10%
<b>Weight evaluation norm</b>	$L_1$ norm

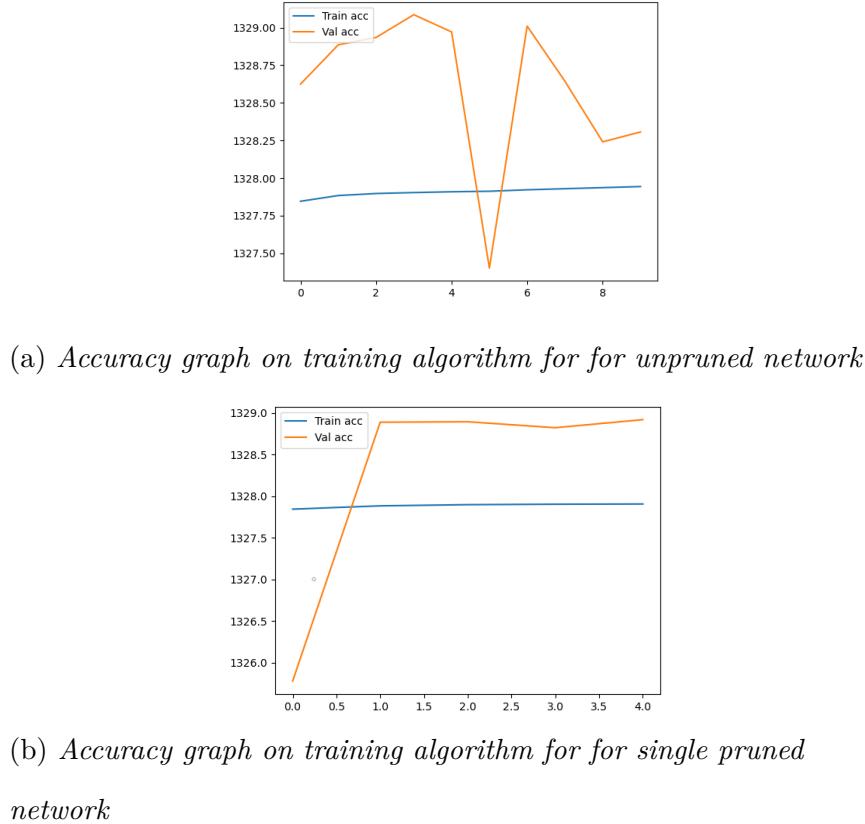
performed, as can be seen in algorithms 1 and 2 for unpruned and single pruned models respectively, as well as iterative algorithms 3 and 4.

Figures 12a and 12b showcase the accuracy plots accross training epochs for both unpruned and single pruned models, respectively. We see that both plots tend to converge to similar accuracy values, give or take random fluctuations. This is to be expected given the similarity of their training algorithms as showcased in algorithms 1 and 2 for unpruned and single pruned models respectively. A similar phenomenon can be witnessed on figures 13a and 13b, for the reasons stated above.

### Evaluating pruned models

The evaluation of pruned models was performed on the following metrics:

- *Parameter metrics* which include **Number of operations**, **Number of parameters** and **Size in disk**.
- *Evaluation metrics* which include **accuracy**, **precision**, **recall**, **f1-score**, and

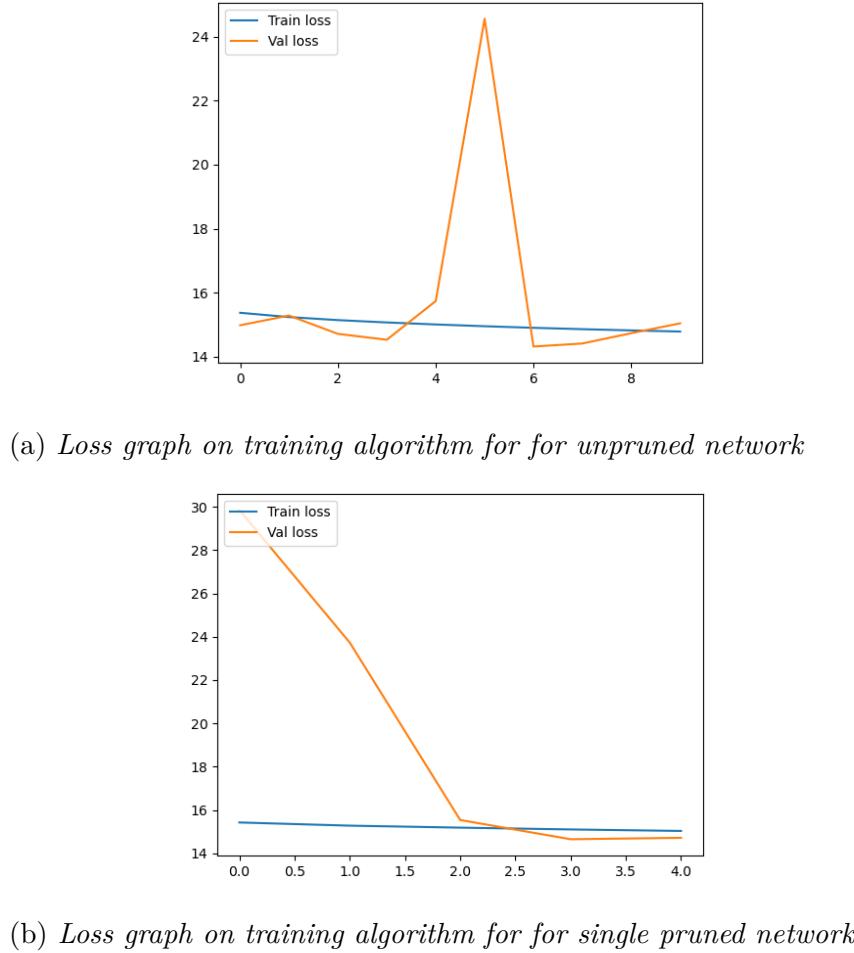
**Figure 12**

Accuracy graphs for training on unpruned and pruned networks

### AUROC.

- *Information flow* which are performed by visualizations of information passing through the networks during a forward pass.

**Parameter metrics.** As seen on table 3, all parameters are equal on pruned and un-pruned networks. This follows from the fact that model pruning sets parameters to 0 (Paganini & Forde, 2020), thus needing to store the value of the parameter, still performing all the same operations, hence the same values on number of operations as well. It should be noted, however, that storing the model using a compression method given the amount of zeroes in the parameters of network should reduce its size in disk.

**Figure 13**

*Loss graphs for training on unpruned and pruned networks*

**Evaluation Metrics.** For the evaluation metrics a transfer learning training was performed on the models to evaluate healthy and non-healthy classes. This normalization allows us to compare models on a real-life setting, as they will be used by medical staff on evaluations. As seen in table 4, models that implement several iterations of pruning methods perform better overall in classification using ChexNet dataset (Jaeger et al., 2014).

**Information Flow metrics.** Measuring information flow of the network requires inspecting the transformations applied to the images throughout, as each input is processed by the different parts. Evaluating the resulting feature maps, as to calculate the activation of each convolution filter. We calculated the overall effect of entire denseblocks

**Table 3**

*Comparison of parameter metrics for Regular Model and Single Pruning Model*

<b>Model</b>	<b># operations</b>	<b># of parameters</b>	<b>Size in Disk</b>
<i>No Pruning</i>	3.7637 GMac, or 7.5274 GFlops	6'968.206	$\approx$ 321.35MB
<i>Single Pruning</i>	3.7637 GMac, or 7.5274 GFlops	6'968.206	$\approx$ 321.35MB
<i>Iterative Pruning</i>	3.7637 GMac, or 7.5274 GFlops	6'968.206	$\approx$ 321.35MB
<i>Incremental Pruning</i>	3.7637 GMac, or 7.5274 GFlops	6'968.206	$\approx$ 321.35MB

(Rajpurkar et al., 2017) by studying the resulting output image given an input on each of the four denseblocks of the architecture. We found that the continuous application of pruning on a model creates redundancies on feature maps (i.e. channels) on the output image. As shown in figures 14, 18, 22, 26, we see that the pruned networks showcase far more redundancies and missing information on the first denseblock of the architecture. These redundancies, given the application of average pooling at the transition layer after each denseblock (Rajpurkar et al., 2017), effectively change the value of the subsequent feature maps, altering activations on deeper layers. Figures 15, 16, and 17 show exactly how these redundancies travel throughout the deeper layers of the unpruned model.

Similarly, deeper feature maps as shown in figures 19, 20, and 21 for the single pruned model showcase that, for the subsequent blocks of the network after the first one, the sparsity of their feature maps decreases drastically. This means that their feature maps, compared to the ones shown in figure 18, have far fewer spots where a map equals 0.

This behaviour can be observed in figures 22, 23, 24, and 29 for the iterative pruning model; and in figures 26, 27, 28, and 26 for the incremental pruning model, respectively.

Analyzing the filters created on the first convolutional layer of the architecture, effectively marking it as the "entry-point" of the model, we see that for pruned models the redundancy of information when updating the parameters of the architecture via backward passing effectively renders some of these filters unnecessary. Figures 30a, 30b, 30c, and 30d

**Table 4**

*Comparison of evaluation metrics for Regular Model and Single Pruning Model*

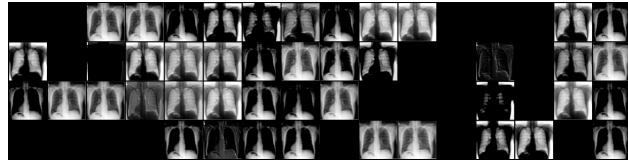
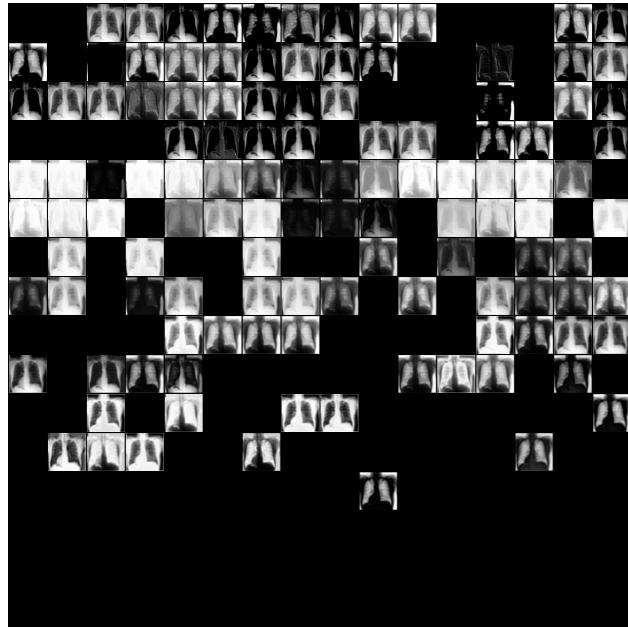
<i>Model</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	<i>Accuracy</i>	<i>Weighted AUROC</i>	<i>Macro AUROC</i>
<i>No Pruning</i>	1.00	0.22	0.37	0.59	0.7022188901901245	0.6894152760505676
	0.53	1.00	0.69			
<i>Single Pruning</i>	1.00	0.22	0.37	0.59	0.6768295764923096	0.6766594052314758
	0.53	1.00	0.69			
<i>Iterative Pruning</i>	1.00	0.22	0.37	0.59	0.7421303391456604	0.7279902696609497
	0.53	1.00	0.69			
<i>Incremental Pruning</i>	1.00	0.22	0.37	0.59	0.7421303391456604	0.7279902696609497
	0.53	1.00	0.69			

show the final state of these filters for the first convolutional layer, showing how some channels can be effectively removed.

Based on these redundancies we can infer that a more in-depth pruning algorithm can be applied (e.g. Structured pruning to remove filters), as information becomes redundant on deep layers within the model. It also shows that shallower layers from the model are also candidates to be included in the pruning algorithm to be applied (Paganini & Forde, 2020), e.g. the convolution layers found at the beginning of the architecture, where the filter maps from figures 30a, 30b, 30c, and 30d were taken. These considerations will be evaluated in future iterations of the application as improvements over the model in question.

### Our proposed application: vIvA-Med Light

vIvA-Med Light is a deep learning model transferred on our organized and curated database to CheXNet. CheXNet is a 121-layer convolutional neural network trained on ChestX-ray 14, currently the largest publicly available chest X-ray dataset, containing over 100,000 frontal-view X-ray images with 14 diseases (Rajpurkar et al., 2017). CheXNet exceeds average radiologist performance on the F1 metric. CheXNet detects all 14 diseases

(a) *Feature maps on input*(b) *Feature maps on output***Figure 14**

*Feature maps on first denseblock for unpruned network*

in ChestX-ray14 and achieves state of the art results on all 14 diseases. CheXNet is a 121-layer Dense Convolutional Network (DenseNet) (Huang et al., 2016) trained on the ChestX-ray 14 dataset. DenseNets improve flow of information and gradients through the network, making the optimization of very deep networks tractable. Researchers replace the final fully connected layer with one that has a single output, after which they apply a sigmoid nonlinearity. (Rajpurkar et al., 2017).

As discussed above, several experiments were performed to fine-tune the existing models and architectures to our particular need. Model reduction, as is specified by works such as (Paganini & Forde, 2020), is needed to ensure a minimal computational print of the

model used, given the reduced pool of resources and the wide variation of them found in modern mobile architectures.

## Results and future work

The metrics for the implementation of vIvA-Med Light on the low-cost app we obtained were organized as per the works of Sendak et al., 2020 and are presented in Table 5. Also, the Receiver Operator Curve (ROC) in test data and the precision recall graphics are depicted in figures 31 and 32 respectively:

<i>Sensitivity</i>	0.987179487179487
<i>Specificity</i>	0.58547008547008506
<i>Precision</i>	0.79875518672199097
<i>Negative Predicted Value</i>	0.96478873239436602
<i>False Positive Rate</i>	0.414529914529914
<i>False Discovery Rate</i>	0.201244813278008
<i>False Negative rate</i>	1.2820512820512799E-2
<i>Accuracy</i>	0.83653846153846101
<i>F1 Score</i>	0.88302752293577902
<i>Matthews Correlation Coeff</i>	0.66124360033285901

**Table 5**

*According to Sendak et al., 2020 results must be provided in such a way that the medical doctors will rely on the tool for decision making*

The architecture of the app followed the traditional mobile-development application architecture of MVVM (Model-View-ViewModel). This facilitated the embedding of an inference model, which could be consumed as an asset for the mobile application, allowing it to compute inferences directly on the mobile application instead of consuming an external service through an internet or intranet connection. This can allow for faster

results and for an element of security since all relevant elements are embedded in the app.

The application prompts the user to select an Rx image from the phone's gallery. If none is found, the user can then take a picture using the camera. After a picture has been selected, the model will process the image and after a few seconds the screen will prompt the result of the inference from the model, stating that the patient from which the Rx was taken was either a *Healthy* individual or an *Unhealthy* one.

Figures 33, 34, 35, and 36 show the current condition of our mobile application, which uses the described learning machines as the core. As can be seen, a medical doctor located in a rural area of an underdeveloped country can easily upload the picture of an X-ray or take a picture and the program will upload and process the image, showing the resulting diagnosis, as shown in figures 34, 35, and 36.

We aimed at developing the least complex model in order to have an inexpensive mobile app in terms of space and processing. The images were classified into healthy and unhealthy, obtaining 2,000 images of non-healthy condition that were, in turn, classified into groups of bacterial pneumonia, viral pneumonia (possible covid19), atelectasis, infiltration, mass, nodule, pneumothorax, edema, emphysema, and fibrosis.

These groups were used to assess the application, providing results of healthy or unhealthy conditions and obtaining successful results when the X-ray image was asymmetric in the lung area. This allowed easy identification of advanced pathological processes and good results in pathologies that did not involve symmetry in the pathological pattern of the pulmonary area.

In relation to suspicious images of COVID19, the machine identified as unhealthy, the patterns that had great pulmonary involvement and, therefore, advanced stages of the disease in terms of early stages with minimal lung involvement, which in most cases is symmetrical, there was greater difficulty in properly identifying it as unhealthy, which led us to generate new processes to strengthen machine learning.

These results lead us to the conclusion that artificial intelligence is essential as a

support tool for medical decision making. It is also useful to further strengthen the practice of medical doctors and specialists in radiology since they already have specific training in making diagnosis by imaging.

General practitioners can greatly benefit from AI. With the help of vIvA-Med Light, they will be able to rely on more accurate diagnosis and, thus, be able to make more accurate decisions regarding the interpretation of results in conjunction with the clinical evaluation of the patient.

Even if we do not have complete certainty of the results, we do have support for medical decision making, which, as the pandemic evolves, can allow for modifications in the protocols for the diagnosis and management of the disease.

At the moment, there is a high probability of ratifying an unhealthy case. In addition, treating doctors may obtain greater clinical support from the evolution of the images that are reported as healthy. Therefore, we can ratify that AI can be a valuable support tool that does not intend to replace medical doctors as the main source in the diagnosis and management of a patient.

However, new questions arise regarding the training of the model and the images to be used. This will require great support from the medical community for the application and re-training of the machine so that it will improve the certainty of the reports of healthy and unhealthy cases.

## Conclusions

During the COVID-19 pandemic, several artificial intelligence tools have been developed by diverse researchers. However, they are far from being credible by medical doctors due to the lack of rigorous clinical studies. In addition, many researchers claim to have the ultimate solution, but neither do they publish their models nor do they satisfy the learning bounds for generalization and robustness of the learning machine. These are not good news if we expect to have a rapid solution for underdeveloped countries.

We consider that we may have developed a new and essential tool, based on artificial intelligence, for the treatment of the pandemic generated by the SARS CoV2 virus. AI is not of recent use in medicine, but the evolution of this technology continues to innovate in many fields, including medicine. Therefore, we need to prepare ourselves to embrace this new tool so that it can render important benefits to the community, in general, and an essential support to medical teams in particular, as we need to include the expertise and domain knowledge of radiologists and medical staff in order to improve upon the applications that use AI that will be developed in the future..

The Latin American health systems, including Colombia's, may collapse due to possible massive visits to the emergency rooms. In rural areas, the scenario is not very optimistic either due to the lack of high level hospitals, imaging devices, and expertise of medical personnel. Looking for a rapid, robust, and usable solution for medical doctors in rural areas, we proposed a mobile device application. The purpose of this application is to provide a support system for medical decision making based on the learning machines trained by our team on a database reliable and curated by both medical doctors and radiologist experts in the field.

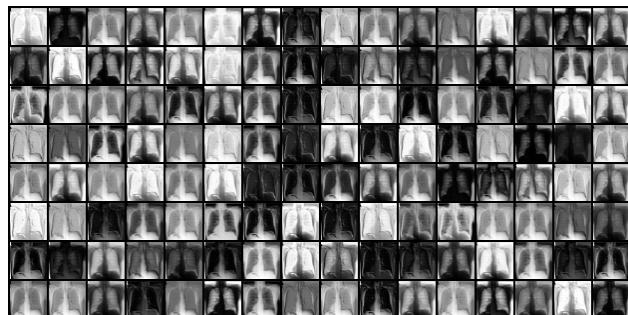
Under developed areas, such as we may find on Colombia's rural regions such as Choco, Guaviare Guainia, Amazonas, and Putumayo, only to mention a few, have little access to medical services and early attention. This tool will result as medical aid to assess other diseases common in vulnerable individuals, such as children or the elderly. Being funded by the state, our solution will be published online at no cost to be used. All our results are supported by clinical trials and metrics on a dataset obtained from colombian facilities by our team of expert radiologists.

The medical responsibility in real life is higher than what scientists can expect and we are not presenting a finished or definite solution. However, our solution is reliable because the learning task can guarantee the learning of at least two classes in our models. We used several databases curated by medical doctors and we were careful in satisfying the

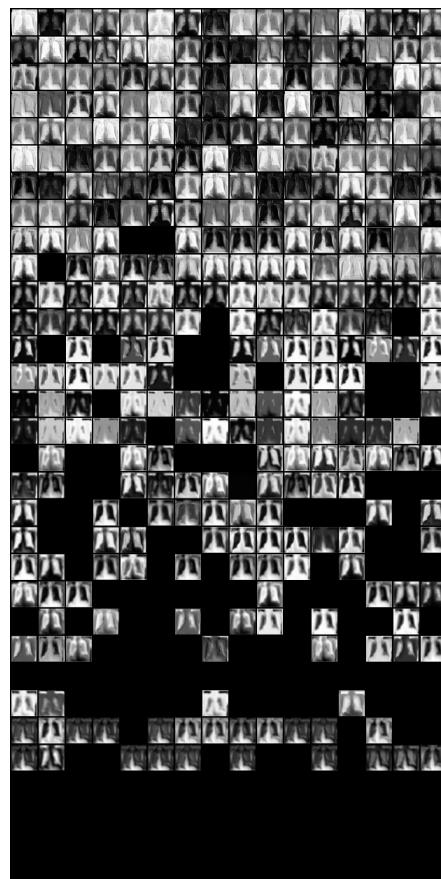
learning theorems for the training, validation, and testing of our models. We did not want to over-fit our models, we wanted to provide a reliable tool. Robustness is necessary as well as providing the results of our model in such a way that they are comprehensible to medical doctors. That is why we presented our solution in terms of the medical labels proposed by Sendak et al., 2020. Our solution was built on a curated database and it is also accompanied by a full solution evaluated retrospectively on a new database built by our medical team and which will be submitted for publication.

### Acknowledgments

We acknowledge the generous help from medical doctors, programmers, and collaborators. We started with no funding for this research, nevertheless funding to continue working in this project was obtained via the Colombian Ministry of Science and Technology, which was also granted to continue working on a solution involving CT Scans.



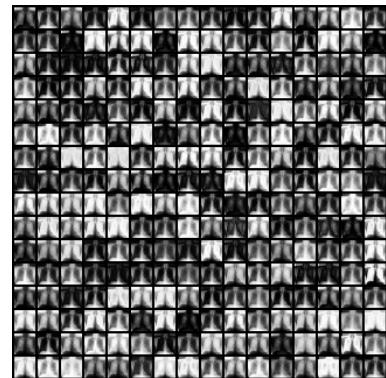
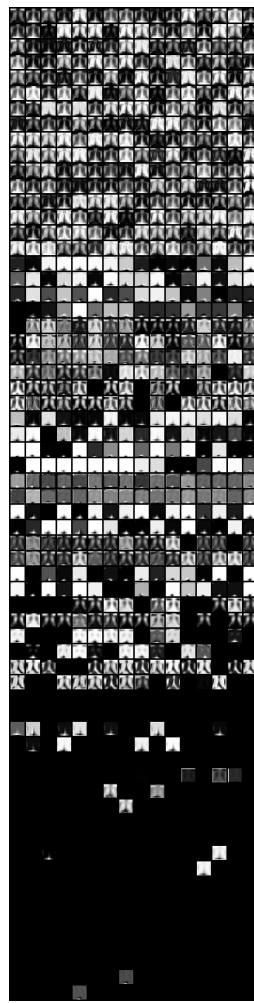
(a) *Feature maps on input*



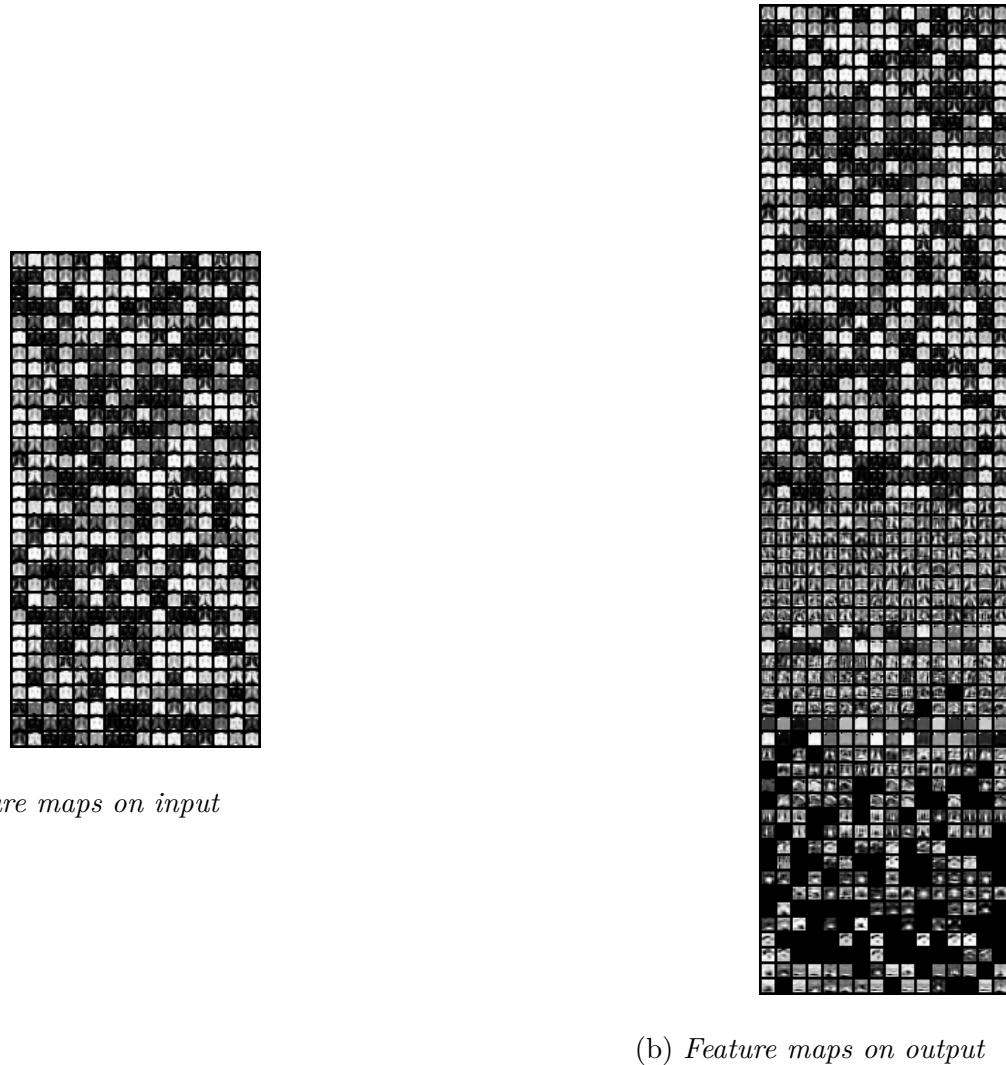
(b) *Feature maps on output*

**Figure 15**

*Feature maps of second denseblock for unpruned network*

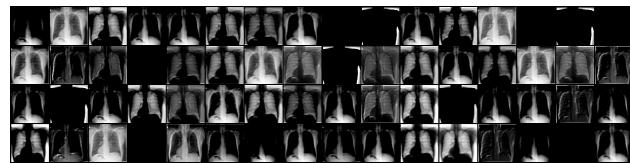
(a) *Feature maps on input*(b) *Feature maps on output***Figure 16**

*Feature maps of third denseblock for unpruned network*

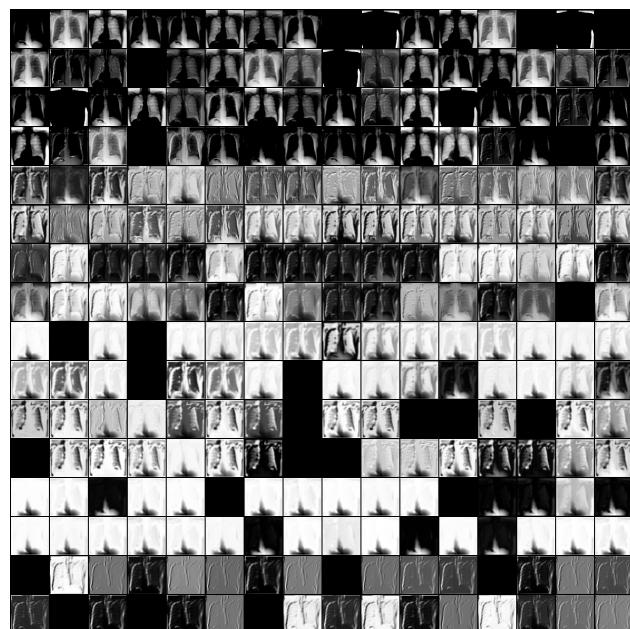


**Figure 17**

*Feature maps of fourth denseblock for unpruned network*



(a) *Feature maps on input*



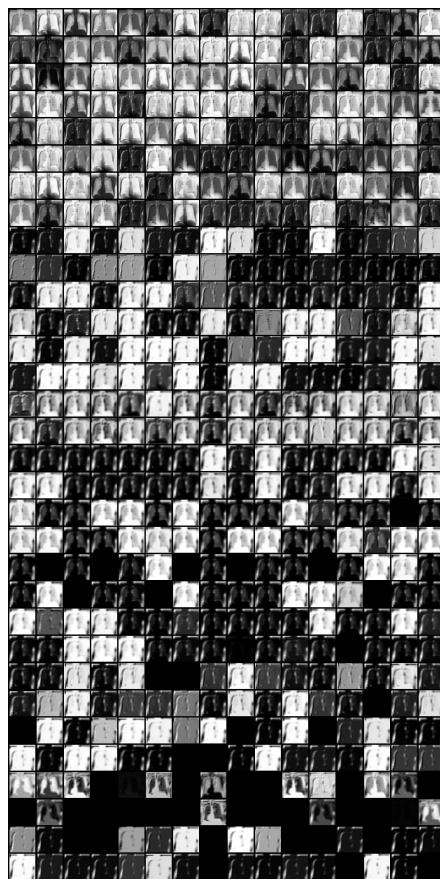
(b) *Feature maps on output*

**Figure 18**

*Feature maps on first denseblock for single pruned network*



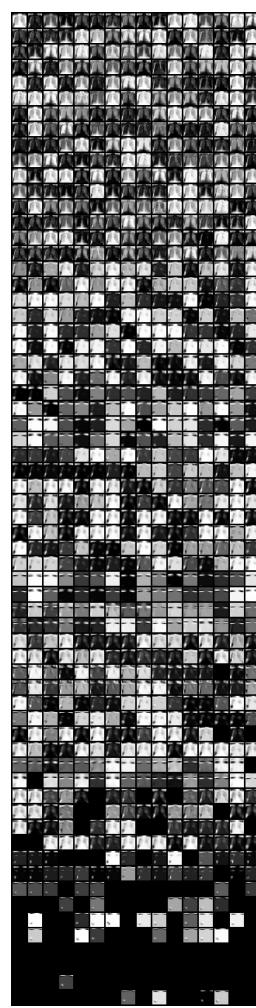
(a) *Feature maps on input*



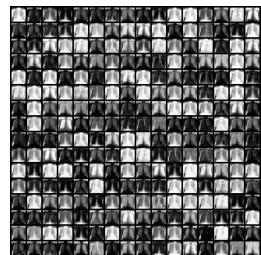
(b) *Feature maps on output*

**Figure 19**

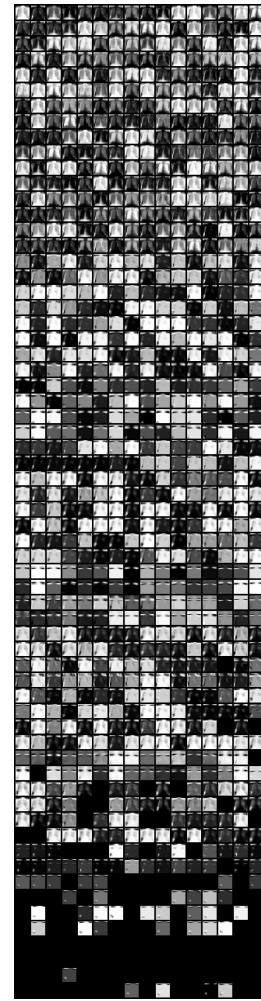
*Feature maps on second denseblock for single pruned network*

(a) *Feature maps on input*(b) *Feature maps on output***Figure 20**

*Feature maps on third denseblock for single pruned network*



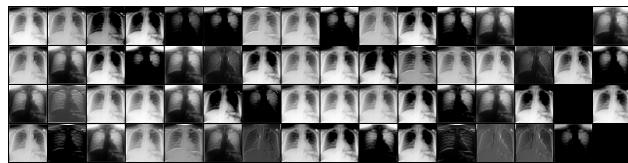
(a) *Feature maps on input*



(b) *Feature maps on output*

**Figure 21**

*Feature maps on fourth denseblock for single pruned network*



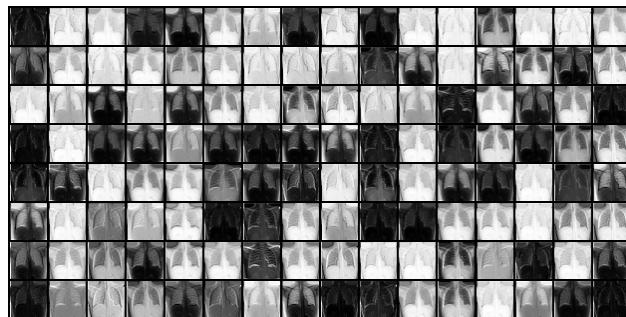
(a) *Feature maps on input*



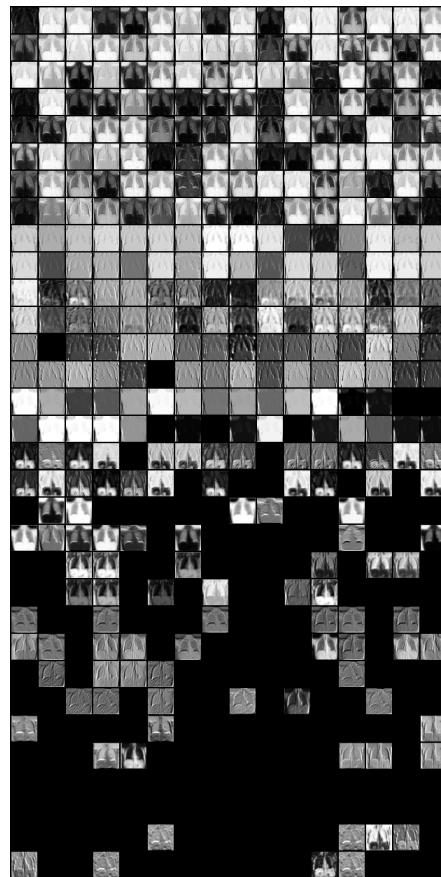
(b) *Feature maps on output*

**Figure 22**

*Feature maps on first denseblock for iterative pruned network*



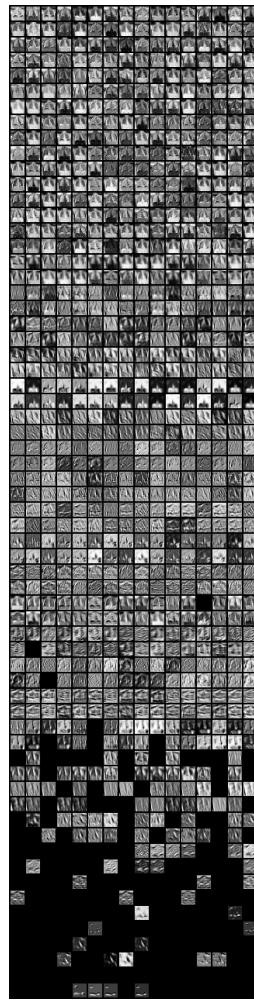
(a) *Feature maps on input*



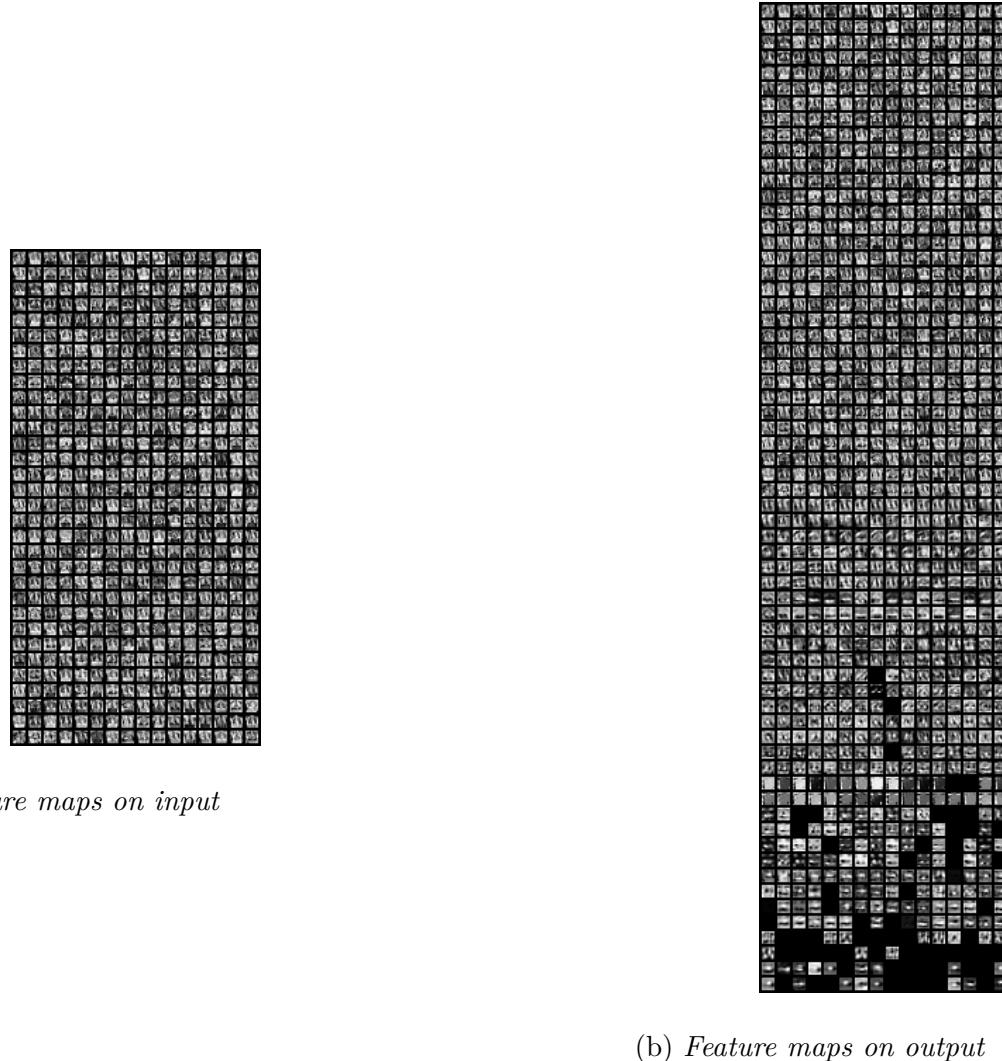
(b) *Feature maps on output*

**Figure 23**

*Feature maps on second denseblock for iterative pruned network*

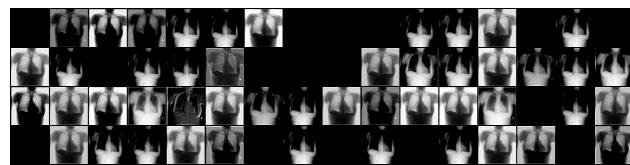
(a) *Feature maps on input*(b) *Feature maps on output***Figure 24**

*Feature maps on third denseblock for iterative pruned network*

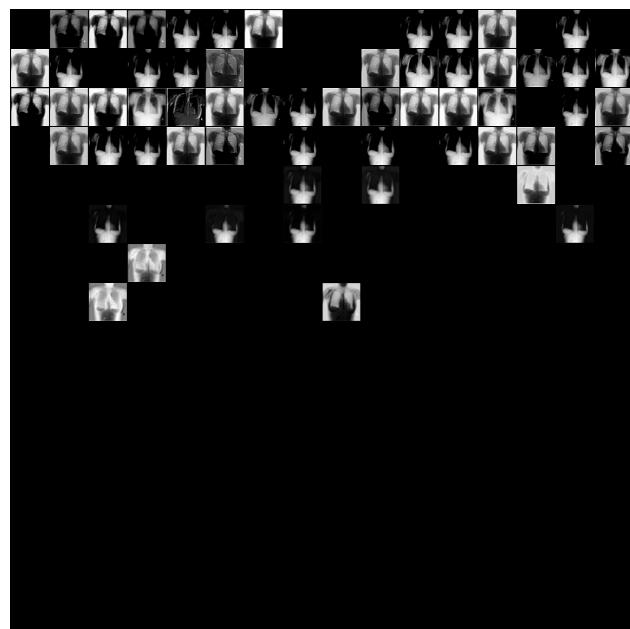


**Figure 25**

*Feature maps on fourth denseblock for iterative pruned network*



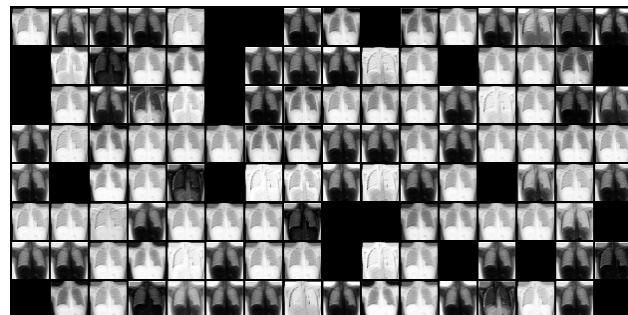
(a) *Feature maps on input*



(b) *Feature maps on output*

**Figure 26**

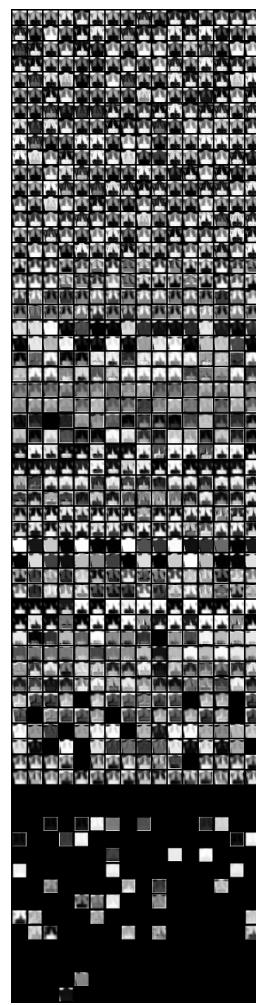
*Feature maps on first denseblock for incremental pruned network*

(a) *Feature maps on input*(b) *Feature maps on output***Figure 27**

*Feature maps on second denseblock for incremental pruned network*



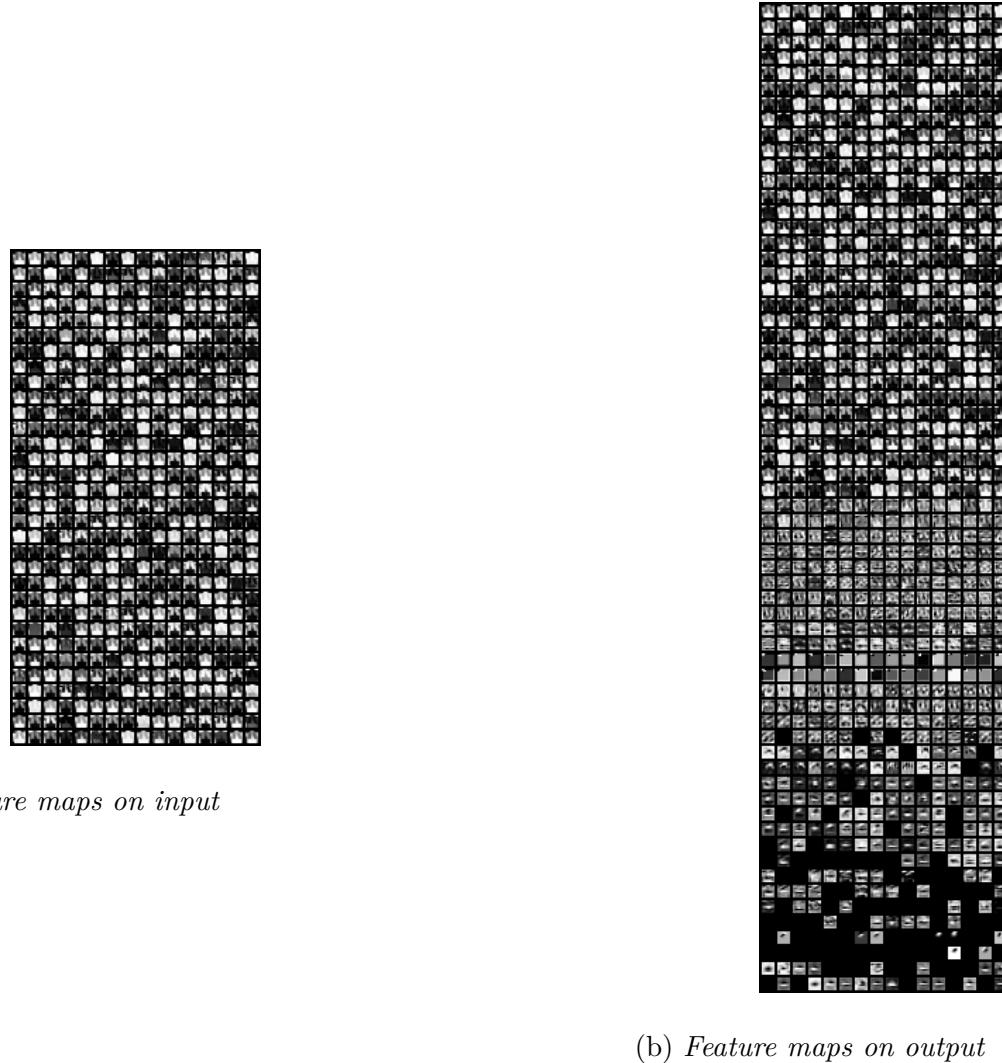
(a) Feature maps on input



(b) Feature maps on output

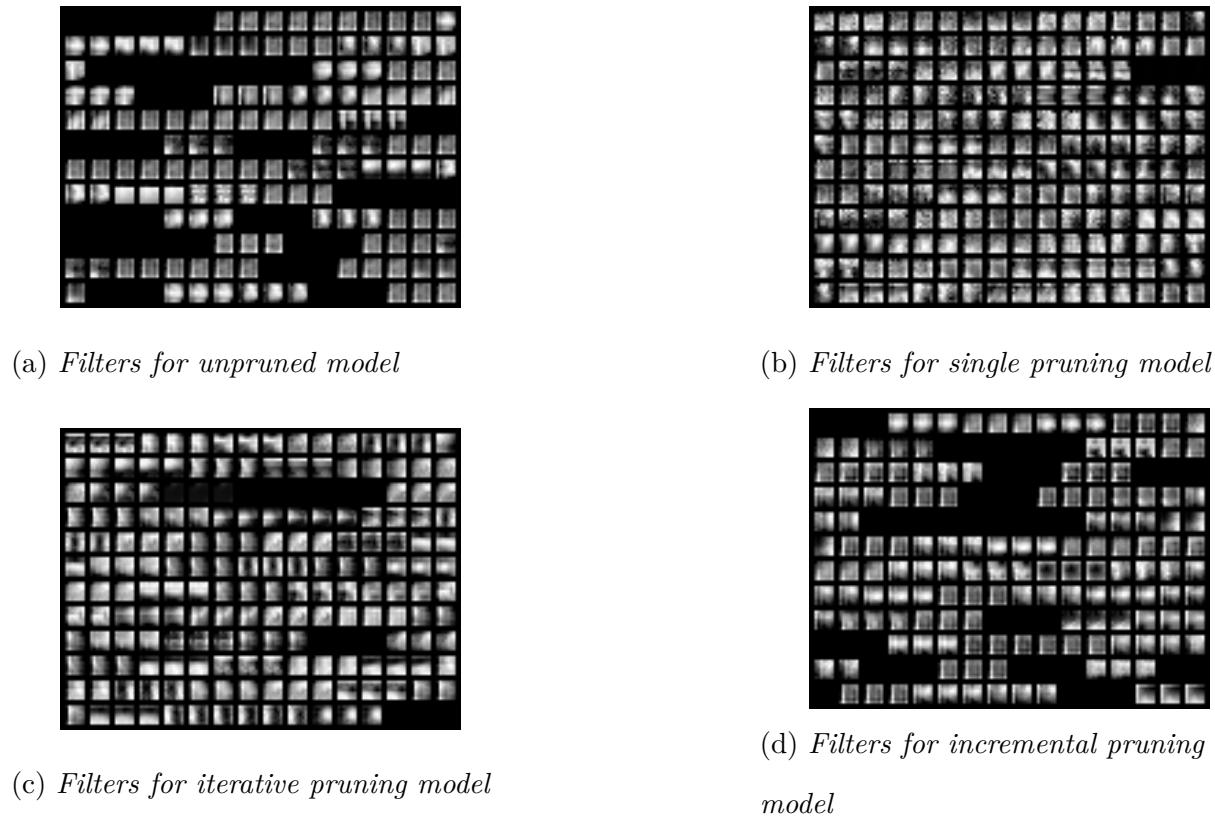
**Figure 28**

*Feature maps on third denseblock for incremental pruned network*

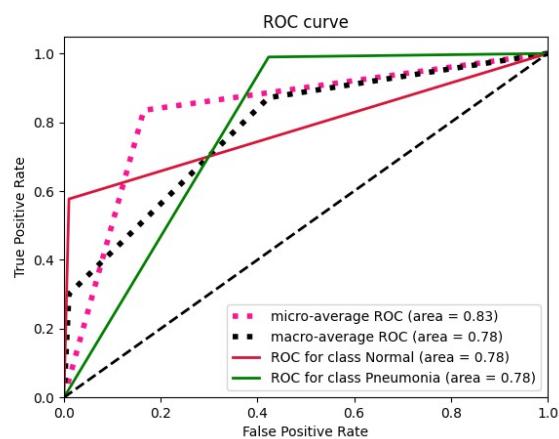


**Figure 29**

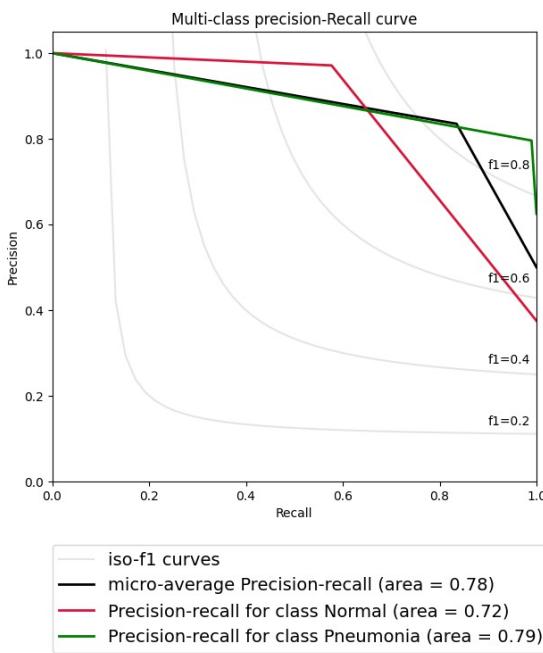
*Feature maps on fourth denseblock for incremental pruned network*

**Figure 30**

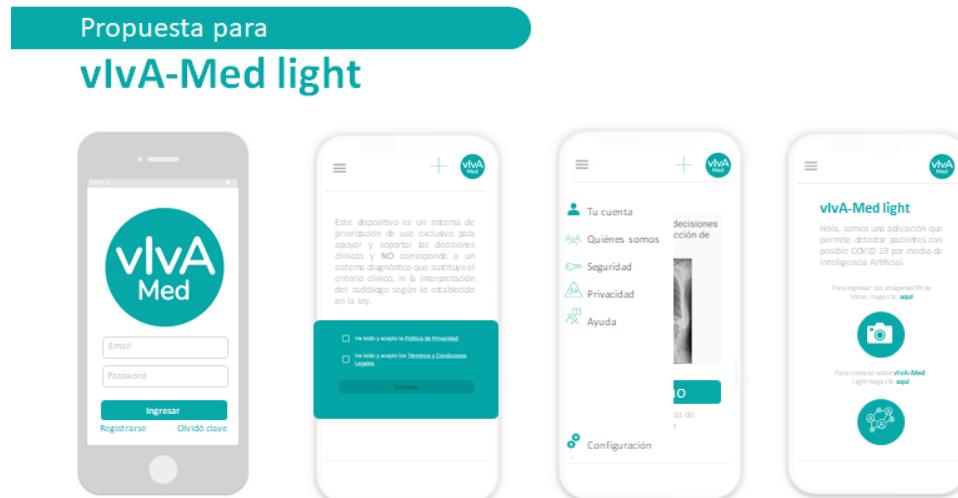
*Filters on first convolutional layer for all architectures*

**Figure 31**

*Receiver Operator Curve in test data for the model trained under the instructions of medical doctors to be used on the low-cost mobile app described in section IV*

**Figure 32**

*Precision and Recall Curve in test data for the model trained under medical doctors instruction to be used on the low-cost mobile app described in section IV*

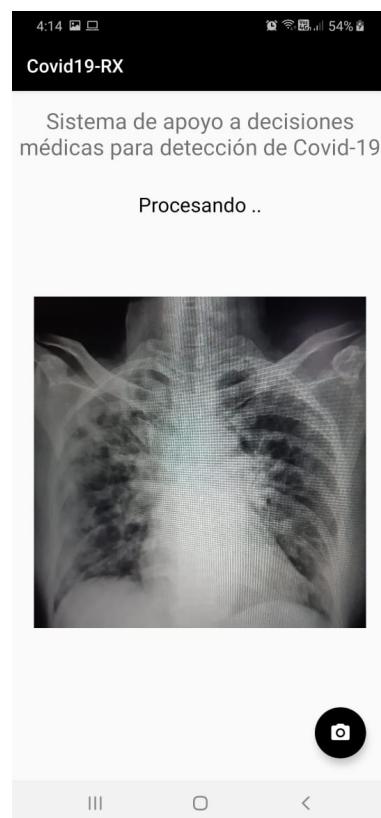
**Figure 33**

*EAFIT University welcome screen to the public mobile application*



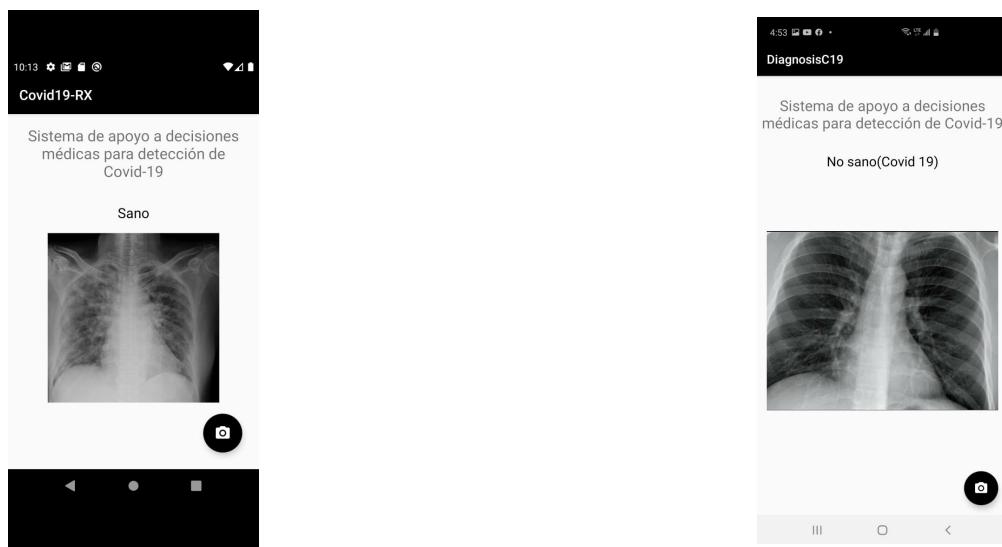
**Figure 34**

*Image loaded to the app*



**Figure 35**

*Processing Image on our application*



(a) Final result on the mobile app for healthy patient

(b) Final result on the mobile app for unhealthy patient

### Figure 36

Final result on the mobile app

## References

- Ajani, T. S., Imoize, A. L., & Atayero, A. A. (2021). An overview of machine learning within embedded and mobile devices—optimizations and applications. *Sensors*, 21(13), 4412.
- Bradford, J., Coe, E., Enomoto, K., & White, M. (2021). Covid-19 and rural communities: Protecting rural lives and health.  
<https://www.mckinsey.com/industries/healthcare-systems-and-services/our-insights/covid-19-and-rural-communities-protecting-rural-lives-and-health>
- Caballero, M. T., Bianchi, A. M., Nuño, A., Ferretti, A. J., Polack, L. M., Remondino, I., Rodriguez, M. G., Orizzonte, L., Vallone, F., Bergel, E., et al. (2019). Mortality associated with acute respiratory infections among children at home. *The Journal of infectious diseases*, 219(3), 358–364.
- Carr, S. (2020). Missed and delayed diagnoses of non-covid conditions – collateral harm from a pandemic. *ImproveDx Newsletter*, 7. <https://www.improvedagnosis.org/improvedx-newsletter/improvedx-july-2020/missed-and-delayed-diagnoses-of-non-covid-conditions-collateral-harm-from-a-pandemic/>
- Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2016). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *CoRR*, *abs/1606.00915*. <http://arxiv.org/abs/1606.00915>
- Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. *CoRR*, *abs/1610.02357*. <http://arxiv.org/abs/1610.02357>
- Cohen, J. P., Bertin, P., & Frappier, V. (2019). Chester: A web delivered locally computed chest x-ray disease prediction system. <https://doi.org/10.48550/ARXIV.1901.11210>
- en Salud, A. V. (2022). Tasa de mortalidad por infección respiratoria aguda (ira) en menores de 5 años - georeferenciado.  
<https://www.asivamosensalud.org/indicadores/enfermedades-transmisibles/tasa-de-mortalidad-por-infeccion-respiratoria-aguda-ira-en>

- Eshaghi, H., Ziae, V., Khodabande, M., Safavi, M., & Memar, E. H. E. (2021). Clinical misdiagnosis of COVID-19 infection with confusing clinical course (M. Dkhil, Ed.). *Case Reports in Infectious Diseases, 2021*, 1–5.  
<https://doi.org/10.1155/2021/6629966>
- Fawcett, T. (2006). An introduction to roc analysis. *Pattern recognition letters, 27*(8), 861–874.
- Frankle, J., & Carbin, M. (2018). The lottery ticket hypothesis: Training pruned neural networks. *CoRR, abs/1803.03635*. <http://arxiv.org/abs/1803.03635>
- Gálvez, M. (2017). Inteligencia artificial en radiología: ¿seremos reemplazados por las máquinas? *Revista chilena de radiología, 23*, 90–90.  
<https://doi.org/10.4067/S0717-93082017000300001>
- Geng, L., & Niu, B. (2022). Pruning convolutional neural networks via filter similarity analysis. *Machine Learning, 1*–20.
- Huang, G., Liu, Z., van der Maaten, L., & Weinberger, K. Q. (2016). Densely connected convolutional networks.
- Jaeger, S., Candemir, S., Antani, S., Yi-Xiáng, J. W., Pu-Xuan, L., & Thoma, G. (2014). Two public chest x-ray datasets for computer-aided screening of pulmonary diseases. *Quant Imaging Med Surg, 4*(6), 475–477.
- Kermany, D. S., Goldbaum, M., Cai, W., Valentim, C. C., Liang, H., Baxter, S. L., McKeown, A., Yang, G., Wu, X., Yan, F., et al. (2018). Identifying medical diagnoses and treatable diseases by image-based deep learning. *Cell, 172*(5), 1122–1131.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems, 1097–1105*.
- Melamed, T. (2018). An active man-in-the-middle attack on bluetooth smart devices. *Safety and Security Studies, 15*, 2018.

Mobile chest x-ray analysis uses machine learning models to demonstrate how smart medical apps can be created. (2019).

<https://www.microsoft.com/en-us/garage/blog/2018/05/mobile-chest-x-ray-analysis-uses-machine-learning-models-to-demonstrate-smart-medical-apps/>

Montoya, O. L. Q., & Paniagua, J. G. (2020). *From artificial intelligence to deep learning in bio-medical applications*. Springer.

Mueller, J. T., McConnell, K., Burow, P. B., Pofahl, K., Merdjanoff, A. A., & Farrell, J. (2021). Impacts of the covid-19 pandemic on rural america. *Proceedings of the National Academy of Sciences*, 118(1), 2019378118.

Organization, P. A. H. (2022). Paho weekly covid-19 epidemiological update - 28 june 2022. *PAHO Weekly COVID-19 Epidemiological Update*.

<https://www.paho.org/en/documents/paho-weekly-covid-19-epidemiological-update-8-february-2022>

Paganini, M., & Forde, J. Z. (2020). On iterative neural network pruning, reinitialization, and the similarity of masks. *CoRR*, abs/2001.05050.

<https://arxiv.org/abs/2001.05050>

Rajpurkar, P., Irvin, J., Zhu, K., Yang, B., Mehta, H., Duan, T., Ding, D., Bagul, A., Langlotz, C., Shpanskaya, K., et al. (2017). Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. *arXiv preprint arXiv:1711.05225*.

Salem, O., Alsubhi, K., Shaafi, A., Gheryani, M., Mehaoua, A., & Boutaba, R. (2021). Man-in-the-middle attack mitigation in internet of medical things. *IEEE Transactions on Industrial Informatics*, 18(3), 2053–2062.

See, A., Luong, M., & Manning, C. D. (2016). Compression of neural machine translation models via pruning. *CoRR*, abs/1606.09274. <http://arxiv.org/abs/1606.09274>

Sendak, M. P., Gao, M., Brajer, N., & Balu, S. (2020). Presenting machine learning model information to clinical end users with model facts labels. *npj Digital Medicine*, 3(1), 1–4.

- Shamout, F. E., Shen, Y., Wu, N., Kaku, A., Park, J., Makino, T., Jastrzebski, S., Witowski, J., Wang, D., Zhang, B., & et al. (2021). An artificial intelligence system for predicting the deterioration of covid-19 patients in the emergency department. *npj Digital Medicine*, 4(1). <https://doi.org/10.1038/s41746-021-00453-0>
- Sohn, H., Tucker, A., Ferguson, O., Gomes, I., & Dowdy, D. (2020). Costing the implementation of public health interventions in resource-limited settings: A conceptual framework. *Implementation Science*, 15(1), 1–8.
- Székely, A., Talanow, R., & Bágyi, P. (2013). Smartphones, tablets and mobile applications for radiology. *European journal of radiology*, 82(5), 829–836.
- Wang, C. J., & Huang, D. J. (2013). The hipaa conundrum in the era of mobile health and communications. *JAMA*, 310(11), 1121–1122.
- Wang, L., & Wong, A. (2020). Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest radiography images.
- Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., & Summers, R. M. (2017). Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2097–2106.
- Wang, Z., Li, C., & Wang, X. (2021). Convolutional neural network pruning with structural redundancy reduction. <https://doi.org/10.48550/ARXIV.2104.03438>
- X-ray app to ease burden on lung specialists and save lives. (2021). *University News*. Retrieved May 6, 2021, from <https://www.gcu.ac.uk/theuniversity/universitynews/2021-x-ray-ai/>
- Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. (2016). Aggregated residual transformations for deep neural networks.
- Yadav, H., Shah, D., Sayed, S., Horton, S., & Schroeder, L. F. (2021). Availability of essential diagnostics in ten low-income and middle-income countries: Results from national health facility surveys. *The Lancet Global Health*, 9(11), e1553–e1560.

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2015). Learning deep features for discriminative localization.