Lung Ultrasound for COVID detection

Pablo Laso-Mielgo [1], A. Ramos-Uparela [1],
Dr. Rafael Garcia Carretero, MD, PhD [2], Dr. Ángel Bueno, MD, PhD [3]

[1] Biomedical Engineering degree, Alcorcón Campus, Universidad Rey Juan Carlos.
[2] Escuela Técnica Superior de Ingeniería en Telecomunicación, Universidad Rey Juan Carlos.
[3] Área hospitalaria, Hospital Universitario Fundación Alcorcón.

Abstract

From the Machine Learning community, US (ultrasound) has not gained as much attention as X-Ray or CT in the context of COVID-19. But many voices from the medical community have advocated for a more prominent role of US in the current pandemic [1]. Moreover, US is a non-ionizing technique, which is increasingly gaining importance as a Point-of-Care technique.

Since two students work on this project, at most points independently, we developed different tools for COVID-19 classification in both US videos and images. Videos may seem more difficult to deal with than images. However, the carry more information that can be used for more efficient classification tools. Additionally, the doctor may provide an image for COVID-19 detection. In this case, two different tools have been developed. One is based on pleural line detection, whereas the other takes strong, pretrained convolutional neural networks, such as InceptionV3 or VGG16. Our model is validated on an external dataset (POCUS).

1 Introduction

This is collaborative project between URJC (Universidad Rey Juan Carlos) and HUFA (Hospital Universitario Fundación Alcorcón). It has been mainly carried out by URJC students Pablo and Alejandro, with the help of Dr. Rafael and Dr. Ángel.

Pablo firstly focused on video preprocessing (HUFA, Butterfly and POCUS dataset) and later joined Alejandro on image classification, which was carried out with several different approaches. Video processing results yield a high accuracy and generalizing ability to new data. On the other hand, image classification required more time, since it was based on video frames, so most are very similar and, even if the accuracy was very high, the generalization to new datasets was slightly lower. However, a large public dataset (POCUS), Transfer Learning, and a thoughtful preprocessing effort allowed us to overcome this challenge and achieve better results.

Transfer Learning performed well on other respiratory pathologies such as pneumonia. For COVID-19 images, however, it did not yield as fine results as video classification, so we also created our own CNN. It is worth mentioning the development of a code able to detect pleural lines.

The goal of this project is processing lung images acquired via US (ultrasound). These images are anatomically poor and, therefore, sometimes difficult to interpret. We can also distinguish between different grades of severity, which are scored in a range of 1-3. although no labels were given for this task.

This project has been motivated by the need of an algorithm that aids physicians to more accurately diagnose diseases, especially COVID-19 [2]. Another major source of motivation is the so-called point-of-care US, that is, provide treatment at the time and place of patient care. There are already portable US equipment. Furthermore, using US as a standard imaging technique (and even rather the so typical stethoscope [3]) would be desirable. Not only would it be faster than other imaging techniques like MRI, but also prevent the patient from receiving high doses of ionizing energy that other techniques, such us CT or PET, produce.

Hence, having an algorithm that helps the medical staff to quickly analyze and score the pathology automatically would be ideal. It would also promote both point-of-care US as a standard diagnosis tool, as well as foster a beneficial transition from ionizing imaging techniques to US, without loss of accuracy or healthcare quality.

2 Problem Statement

This project was initially proposed by Dr. Ángel Bueno (HUFA) as an application. It should quickly identity and score the severity of COVID-19 in different regions of the lungs. The images are acquired by US and are anatomically poor, so preprocessing is probably mandatory.

Lung US (LUS) is used to this aim, which is deemed to be the most effective non-ionizing image modality for this, and also present an alternative [4], or even overcome, CT sensitivity for similar pathologies like pneumonia [5] [6] [7], but also for COVID-19 [8] [9] [10] [11]. As we can

observe in Figure 1, COVID-19 patients present respiratory problems, so LUS can be intelligently used to identify them.

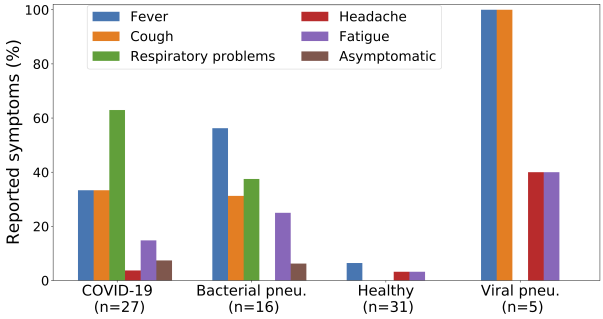

Figure 1. *Lung US Symptoms*

Furthermore, as we can observe in Figure 2, the main characteristics that COVID-19 patients present in LUS images or videos are B-lines and an irregular pleural line, followed by consolidations. A-lines, on the other hand, are idiosyncratic of healthy patients.

All in all, detecting B-lines, and even an irregular pleural line, in a patient is a very solid evidence of COVID-19. Thus, using LUS to address the problem stated in this section is a very promising technique.

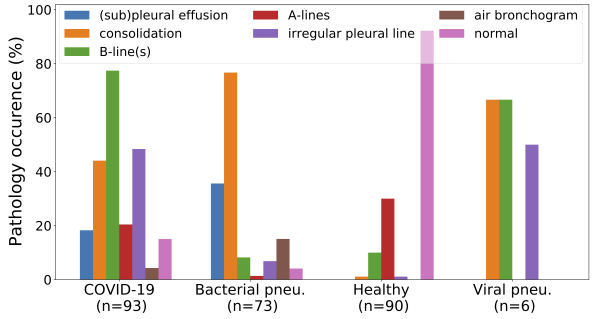

Figure 2. *Lung US Pathologies*

3 Methods

In this Section we describe the procedures followed along our project. These include also attempts that proved to be fruitful, yet not as promising as the last results, described in the next Section of this paper. Overall, we performed several prepossessing steps before applying Deep Learning techniques, which have been considered to be the most effective for this kind of tasks on previous (mostly CT) studies [12] [13] [14].

3.1 Data Acquisition

For the development of this project, specially for the part of training a CNN, the need of a large and labeled database is required. The HUFA kindly helped us in this task, providing us with several images and lung US videos. We were also instructed into the different scores that these disease could manifest [15]. Since the need of medical images and a larger dataset was of paramount importance for the development of the project, we resorted to online accessible databases on this pathology .

Later on, these videos would be divided into frames. Thus, we could deal with images and apply a CNN (based on convolution operations performed upon input images) to our data.

3.2 Data Augmentation

Increasing the database using Data Augmentation was implemented. This can be useful for increasing the number of samples, or for preventing overfitting, by applying random variations to the data. `ImageDataGenerator` [16] allowed us to readily create new data in batches, which also diminished computational burden. Thus, albeit lowering accuracy, we managed to reduce overfitting by creating new data from the original images, mainly by rotation, shifting and zooming.

3.3 POCUS database

As aforementioned, the lack of images to form a quantitatively representative dataset was a major pitfall. To overcome this difficulty that hindered the appropriate training of a CNN, we used the COVID-19 image dataset "POCUS" [1], readily accessible on the Internet.

POCUS database includes both images and videos labelled as regular (healthy), COVID-19, or pneumonia. The dataset (Figure 3) was composed of convex and linear US probes, as well as images and video files. Images of each class are respectively shown in Figure 4.

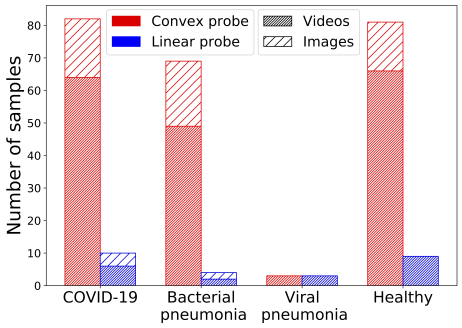

Figure 3. *POCUS dataset size*

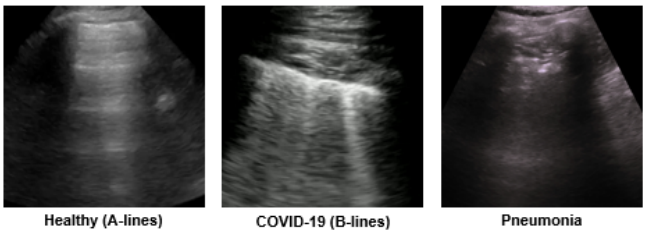

Figure 4. *POCUS image data types*

3.4 Image preprocessing

The main objective of preprocessing our data set was to get much easier for the CNN to better distinguish the proper characteristics of the different pathologies, thus leading with a better performance in terms of classification. The

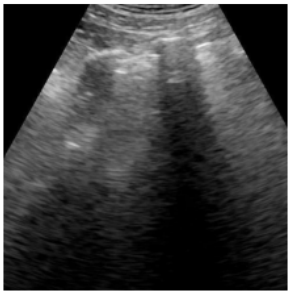

Figure 5. *Original image*

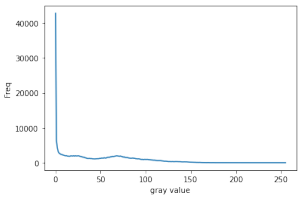

Figure 6. *Original histogram*

process of image preprocessing followed the following steps:

1. **Image histogram equalization**

   A good approach for low contrast images, characterized by a very narrow histogram distribution of gray levels, is to perform a histogram equalization. This technique allows to change the histogram grey values into a more variable distribution, thus giving a higher contrast image.
   This process takes into account the frequency of each grey value and transform it into a probability distribution.The probability of having a certain level of gray,denoted as $p_x(i)$, is the ratio of the number of pixels with the value of gray i with respect to the total pixels of the image.Continued by the calculation of the cumulative distribution function (CDF), defined as

   $$C(i) = \sum_{j=0}^{i} p_x(j) \tag{1}$$

   Where 0 ¡= i ¡= L 1, and L is the total number of possible gray levels in the image.
   Finally, to get the new assigned value of each grey value h(u):

   $$h(u) = round\left(\frac{C(u) - C_{min}}{1 - C_{min}}\right) \tag{2}$$

   Figure above show the histogram and re-arrangement of the resulting transformation (Figures 6 and 7) and the original image (Figures 8 and 5).
   This technique facilitate much more the posterior techniques.
2. **Pleural Line estimation**

   The removal of skin, soft tissue and other elements not involving the lungs(background), is important to facilitate to the CNN a proper pathology(foreground) identification in thorax ultrasound images.

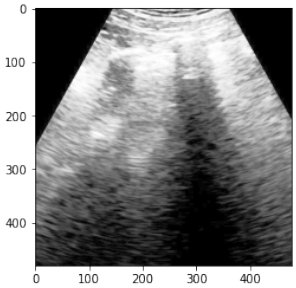

Figure 7. *Equalized image*

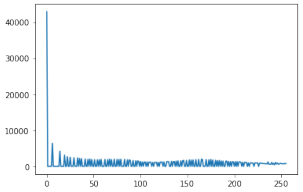

Figure 8. *Equalized histogram*

To remove skin and soft tissue, it is necessary to first identify the pleural line, and for that purpose was following a series of steps:

- Perform a global binarization for each image with the help of the `skimage` library following a certain threshold. This threshold T_0 are determined by minimizing intra-class intensity variance, or equivalently, by maximizing inter-class variance, determined by the ratio:

  $$Q(s) = \frac{V_{inbetween}}{V_{within}} \tag{3}$$

  Each image (I_{ij}) was binarized with this threshold, converting each pixel into an on-pixel or an off-pixel.

  $$B_{ij} = \begin{cases} 0 & \text{if } I_{ij} < T_0 \\ 1 & \text{if } I_{ij} \geq T_0 \end{cases} \tag{4}$$

  Where I_{ij} is the brightness of the pixel.
  The resulting matrix B_{mx1}represents the binarized image on which all future transformations are performed.
- Afterwards, analysis units are set within the binarized image. These are rectangular regions with a height equal to the original image and width of one pixel.

  $$B_{mxn} = \cup_{i=1}^{n} B_{i(mx1)} .... \tag{5}$$

  The total number of on-pixels were calculated in each analysis unit $B_{i(mx1)}$, and use it just for knowing where the middle pointM_i in which the number of on-pixels are equally distributed is, and storing it in a variable X (Figure 8)

  $$X = \left\{\left(M_1\right), \left(M_2\right), \left(M_3\right), ..., \left(M_n\right)\right\} \tag{6}$$

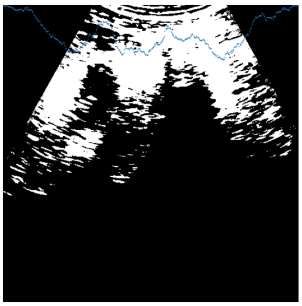

Figure 9. *Thresholding and X points in blue*

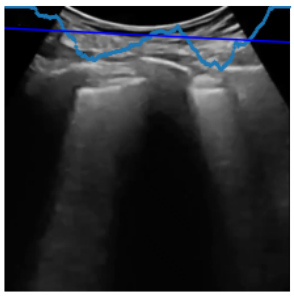

Figure 10. *Regression line of X points*

With these data, we obtained a polygonal profile of the pleural membrane. Because the pleural membrane is usually linear, we made an adjustment of these points to fit a line by minimization of the squared errors. The points of the fitted line are stored in the variable X', where the trace X'is the first approximation of the pleural line (Figure 9)

$$X = \left\{ \left(M_1'\right), \left(M_2'\right), \left(M_3'\right), ..., \left(M_n'\right) \right\} \tag{7}$$

3. **Resection**

   After computing the approximation of the pleural line, we proceed to extract the background information of the original image, getting a resulting image(Figure 13) without the undesired background and perhaps make the feature extraction of the CNN easier.

3.5 Video preprocessing

Apart from preprocessing video frames as independent images, we decided to analyze a video as a whole. This would greatly increase the generalization ability of our classifier,

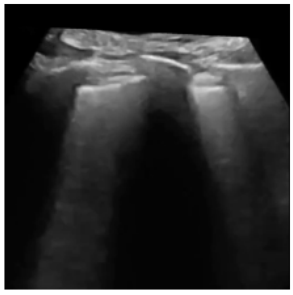

Figure 11. *Final output before resection*

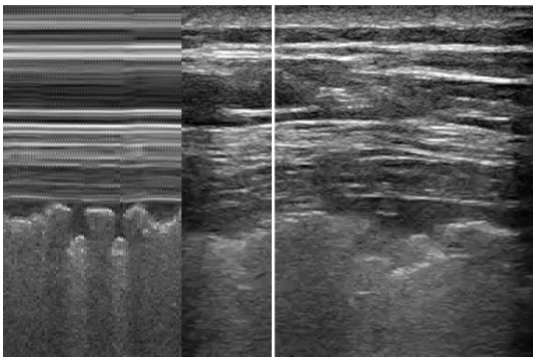

Figure 12. *M-mode US example*

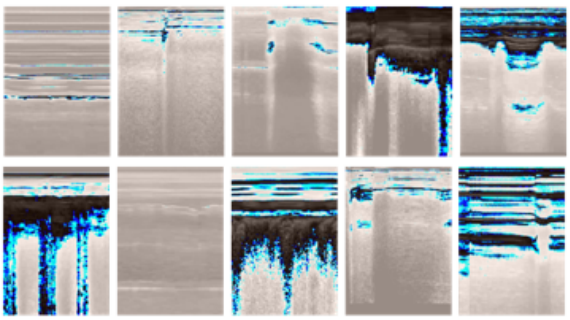

Figure 13. *M-mode images VGG preprocessing*

which was the main challenge we had to cope with.

For such purpose, we created a single image per each independent video in the dataset, by means of converting each US video into an M-mode (Motion mode) US image. As we can see in Figure 12 at the right, a sampling line is taken in the coordinates specified when running the code. Note that we can assess a specific anatomical area by positioning the coordinates in the area of interest. With a sampling line width of 5 pixels, we would take such line from every frame of each video in the dataset. Then, every line would be successively pasted as the column of a new numpy array, giving rise to a new image, as can be observed at the left of Figure 12. Note that long videos will result in wider images, since there exists more frames and, therefore, more columns that the new image will have.

After that, the resulting images would undergo standard preprocessing, including normalization, re-sizing and histogram equalization. Thereafter, VGG preprocessing was used upon them, subsequently obtaining images as shown in Figure 13, which represents one of he batches used for the CNN training.

4 Experiments and Results

This Section includes the results on images (after pleural preprocessing and Transfer Learning) as well as videos.

4.1 Image Classification

As an starting point, without any kind of processing, we just apply some normalization to the images and input it to an standard net, in order to know how the convolutions layers were going to perform.

Several CNN algorithms were designed in an attempt to correctly classify the images. Despite very bad results in the first CNN designs, we eventually came up with a simple design that reached an accuracy of 1.00 in the training set as well as in the validation set, after 10 epochs. The validation set was composed by 57 images (frames) out of 358 altogether, all of them correctly classified into its corresponding score. This result, albeit seemly promising, was however not definitive. Since the images came from three videos (each of them with a different possible score (label)), the images were probably very similar. Furthermore, there existed the possibility that the CNN based its decisions on characteristics of that specific video, such as lightning or intensity, rather than the pathological signs.

In order to be able to contrast these first results we could not but highlight the need for a larger dataset of images, ideally granted by the HUFA; alternatively, obtained from public, online databases on the internet [1]. This dataset, summed up with a the video frames from the corresponding POCUS videos, conform the new dataset used for predictive purposes, with a whole new variability and quantity of acquisitions. The validation set was composed by 653 images out of 3262 images, all of them correctly labelled.

Several architectures were tested, until we designed a simple net (Figure 14) that could give us a clue of how well the data was organized, leading with the clue of possible noisy information in a part of them, the algorithm was being able to highly classify the images, but not to well generalize into a whole new set of unknown images. The results, shown in Figure 15, are very optimistic. We must, however, keep in mind that these results are a consequence of overfitting, so more complex models and further processing is needed.

Layer (type)	Output Shape	Param #
conv2d_9 (Conv2D)	(None, 148, 148, 32)	896
max_pooling2d_9 (MaxPooling2	(None, 74, 74, 32)	0
conv2d_10 (Conv2D)	(None, 72, 72, 64)	18496
max_pooling2d_10 (MaxPooling	(None, 36, 36, 64)	0
conv2d_11 (Conv2D)	(None, 34, 34, 128)	73856
max_pooling2d_11 (MaxPooling	(None, 17, 17, 128)	0
flatten_3 (Flatten)	(None, 36992)	0
dropout_2 (Dropout)	(None, 36992)	0
dense_6 (Dense)	(None, 512)	18940416
dense_7 (Dense)	(None, 3)	1539

Total params: 19,035,203
Trainable params: 19,035,203
Non-trainable params: 0

Figure 14. *First CNN*

4.1.1 Transfer Learning

Instead of starting from scratch, Transfer Learning enables us to use already trained models, albeit they were trained for a different problem. However, we might expect it to behave similarly based on the problem they were originally trained to solve. Transfer learning has the benefit of decreasing the training time for a neural network model and can result in lower generalization error.

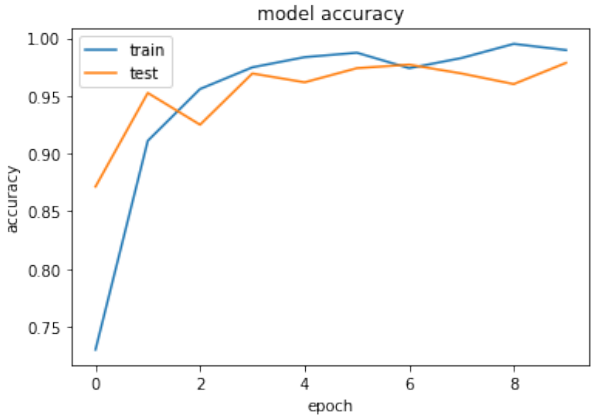

Figure 15. *Model accuracy*

Also, we should bear in mind that (probably) the more similar the original problem (for which the CNN was initially designed) and our challenge are, the more layers (deeper) we will be able to maintain. This is inferred from the fact that the first layers will deal with low-level features, whereas those in the middle shall be much more abstract and complex, fitting to the specific task, problem and dataset they were marginally designed for. Changing just some of these last layers might enough to train an efficient CNN.

Several models from previous studies [17] [18] were tested, among which 'vgg_base', 'vgg_cam', 'mobilenet_v2', 'nasnet', 'dense' and 'resnet50'. All of them had a great tendency to classify images as COVID-19, especially resnet50. The best results were acquired with the following models:

- VGG16
  The VGG16 model was developed by the Visual Graphics Group (VGG) at Oxford and was described in the 2014 paper titled "Very Deep Convolutional Networks for Large-Scale Image Recognition". Its structure is described in Figure 16. By default, the model expects color input images to be re-scaled to the size of 224×224 squares. It can be loaded `from keras.applications.vgg16` with `import VGG16` [16]. From here on, several parameters may be re-trained. Thus, we were able to get an accuracy of 0.69 in the validation set. However, we noted, by plotting the confusion matrix, that some errors were regular cases incorrectly classified as COVID-19. However, both COVID-19 and pneumonia cases were correctly classified.
- InceptionV3
  InceptionV3 is a convolutional neural network for assisting in image analysis and object detection, and got its start as a module for Googlenet. Its architecture is shown as a simplified image in Figure 17. It is therefore more likely to yield relevant answers to our specific problem. Furthermore its has already been implemented in a myriad of cases, some of them yielding expert-level diagnostic accuracy and been proven more efficient than other previous techniques in de-

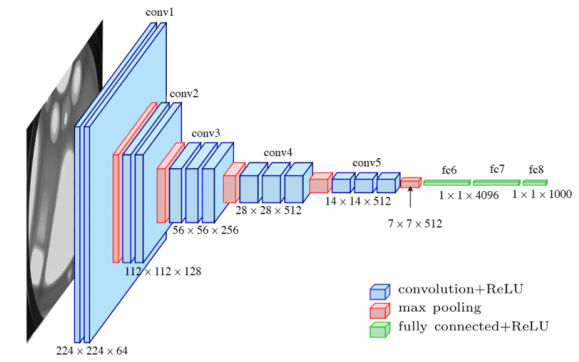

Figure 16. *VGG16 structure*

tecting B-lines, merged B-lines, lack of lung sliding, consolidation and pleural effusion [19]. In this case, the accuracy for the three is slightly lower (0.58), so we can assert that VGG performs better. Furthermore, just like the former VGG model, choosing more than two epochs affect negatively the performance of the model.

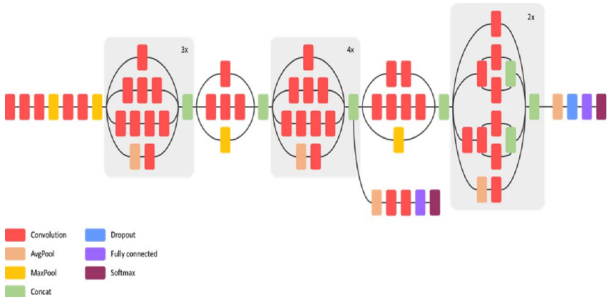

Figure 17. *InceptionV3 structure*

For both cases we constructed a head of a new model that would be placed on top of the base model. Thus, we train the last layers such that it could adapt to our specific case, while also using a pretrained and effective model, which led to a smaller computational burden, overall.

Both models flagrantly showed an unwanted tendency from the ”regular” to the ”COVID” class, as shown in Figures 18 and 19. However, these results, albeit unwanted, are not so bad among all the possible incorrect answers we could have encountered. This is because we would rather have a normal patient diagnosed as COVID-19 than the opposite, that is, it has a higher cost. Armed with this reasoning, we should look at the Recall (TP/(TP+FN)) measured for ”COVID” and ”pneumonia” in Tables 1 and 2 is, since the cost of a False Negative is higher (we want to avoid at all costs having infected people (Actual Positives) in contact with others). Contrarily, we should look closer at the Precision (TP/(TP+FP) measured for ”normal”, since the cost of a False Positive can be, just like before, undesirable for society. We can also use the F1-score (which takes into account both precision and recall) for the ”normal” class. Hence, we can readily observe that VGG performs somewhat better than Inception in this first attempt. However, the results are very humble. Since adding epochs do not contribute to a better performance, we conclude from these results that further preprocessing of our images is needed, as well as trying other layers to re-train, or choosing another head to add at the end of the architecture to better adapt to our task.

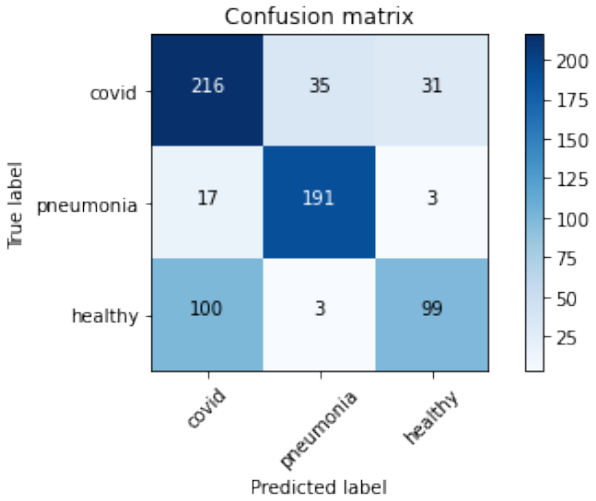

Figure 18. *VGG16 Confusion Matrix*

VGG16 metrics			
class	Precision	Recall	F1
COVID-19	0.65	0.77	0.70
Pneumonia	0.83	0.91	0.87
Regular	0.74	0.49	0.59

Table 1. *VGG16 metrics*

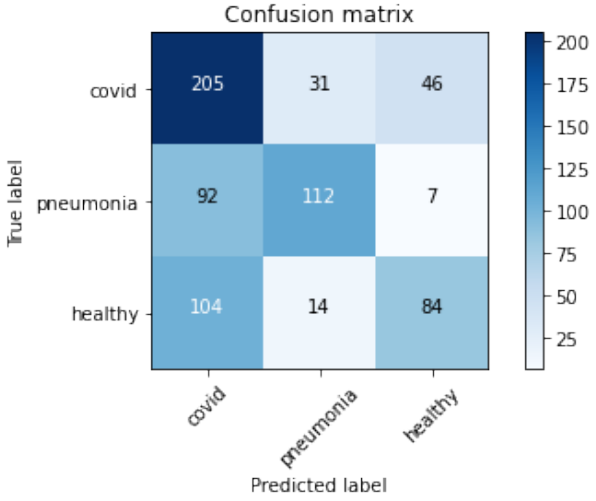

Figure 19. *InceptionV3 Confusion Matrix*

InceptionV3 metrics			
class	Precision	Recall	F1
COVID-19	0.51	0.73	0.60
Pneumonia	0.71	0.53	0.61
Regular	0.61	0.42	0.50

Table 2. *InceptionV3 metrics*

4.1.2 Complex CNN

After performing a much more complex pre-processing stage (described in previous sections) than initially, we de-

cided to use the same CNN architecture(See Figure 15) to compare results. This time , however, we had much more images and a strong pre-processing that showed significantly more promising results, since the generalization adaptability was now stronger as we get rid of noisy information. Nevertheless,the image dataset has been reduced by the little computational capacity with which we had, seeing the set reduced by 50 images per category. Anyway, to try to appease this reduction, we apply a Data augmentation method in order to have a more robust set (Procedure showed in Figure 20)

```
train_datagen = ImageDataGenerator(
    rescale = 1./255.,
    rotation_range=40,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
    )

train_generator = train_datagen.flow_from_directory(TRAINING_DIR,
                                                    batch_size=10,
                                                    class_mode='categorical',
                                                    target_size=(150, 150))
```

Figure 20. *Data augmentation*

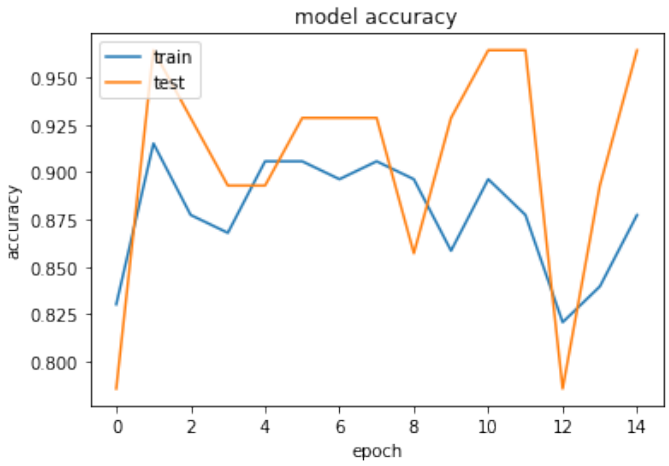

Figure 21. *Loss/Accuracy Final CNN*

As shown in Figure 21 , the capability of the model generalization of new incoming data has been improved.

4.2 Video Classification

Finally, after video preprocessing had been carried out, images had multiple sizes. Therefore, they were resized to the mean height and width, that is, (723, 512). We decided to use the mean rather than the maximum, as usually is done, because images were of all kind of sizes. This is because their size depends on the video duration. We consider it simpler to just use the mean, so that the least interpolation possible was performed overall.

A very simple CNN was able to classify these images (one per video) with an accuracy greater than 0.9. As schematically shown in Figure 22, the CNN consisted on two initial kernel layers. The first layer consisting of 32 filters, whereas the second one, 64 feature maps. All kernels had a size of 3x3, a stride of 2, and were activated or not according to a RELU activation function. These kernel layers were followed by 2x2 maxpooling layers, so the output of each was twice smaller. Thereafter, a dense layer was designed along with a softmax which was used at the end of it, so that we could convert the final output value into probabilities, for labelling.

We used 75% for the training set, 15% for validation, and the rest (15%) for testing. 72 independent videos and from different sources were used as dataset in the last attempt. With random sampling and a batch size of 10, the folder for the two classes (COVID-19 and healthy) were created. By random sampling, we also doubled the initial size per set. We chose *categorical_crossentropy* for loss evaluation, and *accuracy* as the metric to evaluate performance. Additionally, a learning rate of 0.0001 proved to be the best option, since it was not too small to get stuck in a local minimum, but small enough to converge towards a very good result. Since the learning rate was small and the videos in the dataset were of all kinds and different and utterly independent sources, the CNN would take 8 epochs to finally reach a 90% of validation accuracy. Henceforward, the training accuracy improved constantly, but not so the validation accuracy, which started fluctuating. Thus, we deem it to be convenient to perform early stopping at epoch 8. Otherwise, we would risk over-fitting our model.

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_2 (Conv2D)            (None, 723, 512, 32)      896
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 361, 256, 32)      0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 361, 256, 64)      18496
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 180, 128, 64)      0
_________________________________________________________________
flatten_1 (Flatten)          (None, 1474560)           0
_________________________________________________________________
dense_1 (Dense)              (None, 2)                 2949122
=================================================================
Total params: 2,968,514
Trainable params: 2,968,514
Non-trainable params: 0
_________________________________________________________________
```

Figure 22. *video CNN architecture*

Finally, we used the testing set (20 images) to ratify our previous assumptions. The results are shown in Figure 23 as a confusion matrix. The results reflected the same as the validation set. We can observe that all COVID-19 cases are classified as such. This is crucial, since we do not want to have an unknown and unidentified COVID-19 vector, which can spread the disease and lead to more patients. On the other hand, almost all normal (healthy) cases were classified as non-COVID19.

Taking everything into consideration, we can claim that, even if the videos were from three different independent sources (increasing variability), a thoughtful and complex preprocessing of the videos (turning them into an image per video) allowed for a very simple CNN to achieve a 90% accuracy in both the validation and training set, added to a great capability of generalization (preventing overfitting).

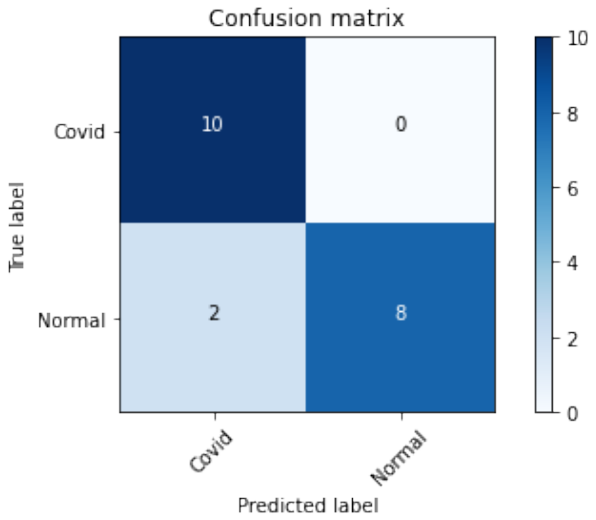

Figure 23. *Confusion Matrix for video classification*

5 Conclusion

After several different attempts for image classification, we conclude that our preprocessing methods allows both for a higher accuracy, as well as greater generalizing ability, so that a simple CNN can readily identify COVID-19 images. Similarly, Transfer Learning, specifically VGG16, has also proven to be effective for classification of pneumonia, COVID-19 and healthy patients, with an accuracy of 0.73%. Interestingly, it is a fairly effective option for pneumonia detection (0.83 precision, 0.91 recall, 0.87 F1; for pneumonia only), similar to previous studies (0.83 accuracy) [20]. However, in the case of COVID-19, which is the main goal of this paper, has worse results and, therefore, we discarded it in search of more efficient options for COVID-19 detection.

Video processing has rendered the most promising results (90% validation accuracy). A very simple and basic CNN architecture already shows a great ability to generalize to three independent datasets of videos, while maintaining a high accuracy that surpasses previous works on this problem [20]. Moreover, and unlike image processing, the high accuracy in video classification shows no over-fitting, since all images (one per video) were independent of each other. Furthermore, its preprocessing allows for a more rapid, and computationally cheap, classification.

6 Future lines

Both image and video classifiers can be tried in clinical practice for further enhancement. Image classification showed a lower generalizing ability at first. However, the POCUS database offered a large set of images and the preprocessing carried out were finally able to increase its generalizing ability and showed an acceptable accuracy without compromising the model to overfitting. Since the last results are better, we strongly suggest making a more complex CNN, as well as increasing such database with many other videos, so that there are much more different frames and the image classifier can be further enhanced based on our preprocessing methods and CNN architecture.

On the other hand, the video classifier achieved a high accuracy and was able to generalize, since there were many different videos from independent sources. All video processing was done by student Pablo's using a very modest Personal Computer. We believe, nonetheless, that more complex CNN architectures should be tried out in more powerful devices -and an even higher accuracy would be attained. Moreover, further pre-processing could be tested, although we do recommend maintaining the core idea of our pre-processing methods, since it was precisely M-mode US and VGG pre-processing what made it possible, and boosted video classification. In terms of image processing we faced the same problem. The resources that we had were not able to handle a huge amount of images and complex preprocessing, so the training of the CNN was limited, as we need to select an specific batch of each category folder that our PC's could be able to work with.

It is important also to state the simplicity of our approach to pleural resection. We believe that applying a more effective way to get the same or even better result would be optimal, and a way to accomplish that is perhaps building a segmentation CNN able to determine more accurately the pleural position and get rid of the actual background points of the image without even loosing important information.

As a further scope we could face this problem by simply using powerful resources as Amazon Web Services(AWS) or Google Cloud Platform(GCP), just setting a sufficient number of virtual machines with standard CPU power to help us process the whole amount of POCUS images.

Workload and Acknowledgments

Special thanks to Dr. Ángel (HUFA), LAIMBIO staff (URJC), Dr. Rafael, and Prof. Cristina, Ph.D., for their technical, and sometimes personal, support throughout the whole project.

The paper was written jointly. Each student wrote the following sections:

- **Pablo** sections Abstract, 1, 2, 3.1, 3.3, 3.5, 4.1.1, 4.2, 5, 6 and References.
- **Alejandro** sections: 3.4,4.1, 4.1.2, 5, 6.

As for the practical part, there were several collaborators:

- **HUFA**: Dr. Ángel instructed us on LUS images.
- **Dr. Rafael** was very helpful and supportive in the first stages of the development of this project.
- **LAIMBIO**: Both Norberto and David Viar helped us thanks to their previous work (TFG, or Bachelor's degree thesis on pneumonia detection, also in HUFA.)
- **Student Pablo**: First CNN (overfitting). POCUS dataset management: organizing images into folders (labels), creating frames from the videos, standard preprocessing, creating splits for cross-validation. Standard preprocessing for images (normalization, histogram equalization and mean filter). Transfer Learning (unexpected, yet promising results for pneumonia). Video preprocessing (standard preprocess-

ing, resizing, VGG preprocessing, M-mode US). Video classification (simple CNN, high accuracy and generalizing ability).

- **Student Alejandro**: Complex image preprocessing (pleural line detection). Complex CNN (overfitting).

References

[1] J. Born, N. Wiedemann, M. Cossio, C. Buhre, G. Brändle, K. Leidermann, A. Aujayeb, M. Moor, B. Rieck, and K. Borgwardt, "Accelerating detection of lung pathologies with explainable ultrasound image analysis," *Applied Sciences*, vol. 11, p. 672, Jan 2021.

[2] "Pagano a, numis fg, visone g, pirozzi c, masarone m, olibet m, nasti r, schiraldi f, paladino f. lung ultrasound for diagnosis of pneumonia in emergency department. intern emerg med. 2015 oct;10(7):851-4. doi: 10.1007/s11739-015-1297-2. epub 2015 sep 7. pmid: 26345533.,"

[3] "Buonsenso d, pata d, chiaretti a. covid-19 outbreak: less stethoscope, more ultrasound. lancet respir med. 2020 may;8(5):e27. doi: 10.1016/s2213-2600(20)30120-x. epub 2020 mar 20. pmid: 32203708; pmcid: Pmc7104316.."

[4] "Fiala mj. ultrasound in covid-19: a timeline of ultrasound findings in relation to ct. clin radiol. 2020 jul;75(7):553-554. doi: 10.1016/j.crad.2020.04.003. epub 2020 apr 18. pmid: 32331781; pmcid: Pmc7165267..,"

[5] "Amatya y, rupp j, russell fm, saunders j, bales b, house dr. diagnostic use of lung ultrasound compared to chest radiograph for suspected pneumonia in a resource-limited setting. int j emerg med. 2018 mar 12;11(1):8. doi: 10.1186/s12245-018-0170-2. pmid: 29527652; pmcid: Pmc5845910.,"

[6] "Lichtenstein d, goldstein i, mourgeon e, cluzel p, grenier p, rouby jj. comparative diagnostic performances of auscultation, chest radiography, and lung ultrasonography in acute respiratory distress syndrome. anesthesiology. 2004 jan;100(1):9-15. doi: 10.1097/00000542-200401000-00006. pmid: 14695718.,"

[7] "Bourcier je, braga s, garnier d. lung ultrasound will soon replace chest radiography in the diagnosis of acute community-acquired pneumonia. curr infect dis rep. 2016 dec;18(12):43. doi: 10.1007/s11908-016-0550-9. pmid: 27785748.,"

[8] "Antùnez-montes oy, buonsenso d, paz-ortega so. rationale for the routine application of lung ultrasound in the management of coronavirus disease 2019 (covid-19) patients in middle- to low-income countries. ultrasound med biol. 2020 sep;46(9):2572-2574. doi: 10.1016/j.ultrasmedbio.2020.05.020. epub 2020 jun 5. pmid: 32593499; pmcid: Pmc7274635.,"

[9] "Ragnoli b, malerba m. focus on the potential role of lung ultrasound in covid-19 pandemic: What more to do? int j environ res public health. 2020 nov 13;17(22):8398. doi: 10.3390/ijerph17228398. pmid: 33202769; pmcid: Pmc7698284.,"

[10] "Volpicelli g, lamorte a, villén t. what's new in lung ultrasound during the covid-19 pandemic. intensive care med. 2020 jul;46(7):1445-1448. doi: 10.1007/s00134-020-06048-9. epub 2020 may 4. pmid: 32367169; pmcid: Pmc7196717.,"

[11] "Kulkarni s, down b, jha s. point-of-care lung ultrasound in intensive care during the covid-19 pandemic. clin radiol. 2020 sep;75(9):710.e1-710.e4. doi: 10.1016/j.crad.2020.05.001. epub 2020 may 13. pmid: 32405081; pmcid: Pmc7218373.,"

[12] S. Ying, S. Zheng, L. Li, X. Zhang, X. Zhang, Z. Huang, J. Chen, H. Zhao, R. Wang, Y. Chong, J. Shen, Y. Zha, and Y. Yang, "Deep learning enables accurate diagnosis of novel coronavirus (covid-19) with ct images," 2020.

[13] "Hui, d. s. et al. the continuing covid-19 epidemic threat of novel coronaviruses to global health-the latest 2019 novel coronavirus outbreak in wuhan, china. int. j. infect. dis. 91, 264–266 (2020).,"

[14] M. Z. Alom, M. M. S. Rahman, M. S. Nasrin, T. M. Taha, and V. K. Asari, "Covid_mtnet: Covid-19 detection with multi-task deep learning approaches," 2020.

[15] "Smith mj, hayward sa, innes sm, miller asc. point-of-care lung ultrasound in patients with covid-19 - a narrative review. anaesthesia. 2020 aug;75(8):1096-1104. doi: 10.1111/anae.15082. epub 2020 apr 28. pmid: 32275766; pmcid: Pmc7262296.,"

[16] F. Chollet, "keras." `https://github.com/fchollet/keras`, 2015.

[17] M. Farooq and A. Hafeez, "Covid-resnet: A deep learning framework for screening of covid19 from radiographs," 2020.

[18] M. Loey, F. Smarandache, and N. E. Khalifa, "Within the lack of chest covid-19 x-ray dataset: A novel detection model based on gan and deep transfer learning," *Symmetry*, vol. 12, p. 651, 04 2020.

[19] S. Kulhare, X. Zheng, C. Mehanian, C. Gregory, M.-H. Zhu, K. Gregory, H. Xie, J. Jones, and B. Wilson, *Ultrasound-Based Detection of Lung Abnormalities Using Single Shot Detection Convolutional Neural Networks: International Workshops, POCUS 2018, BIVPCS 2018, CuRIOUS 2018, and CPM 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16–20, 2018, Proceedings*, pp. 65–73. 09 2018.

[20] "David viar hernández and norberto malpica gonzález. .detección automática de neumonía en pediatría usando ultrasonidos."