FDA SUBMISSION: CXR PNEUMONIA DETECTOR

Luigi Saetta

LSAETTA MEDICAL DEVICES Rome, Italy

Last update: 09/01/2021

Table of Contents

Algorithm Description.

1. General Information.

Introduction:

The device we're submitting for clearance is a **Software as a Medical Device (SAMD**): a software application based on a Deep Learning Model, using a Convolutional Neural Network (CNN).

Intended Use Statement:

The proposed software application is intended to assist radiologists in rapid identification of pneumonia in frontal Chest X-Rays (CXR).

Indications for Use:

- The model can be used to **analyze frontal Chest X-Rays**, supplied as **DICOM (DCM)** files.
- The model has been trained and therefore can be safely used on CXR with **Anterior-Posterior (AP) and Posterior-Anterior (PA)** views.
- It is recommended to use the model only with images from patients with **age between 10 and 80 years old.**
- The model can be used in Hospital to **prioritize review by radiologists** and give higher priority to CXR showing signs of pneumonia.
- If the model predicts that **a CXR is positive**, it **should be placed in the priority queue** for Pneumonia diagnosis; If the model predicts that the **CXR is negative**, this **doesn't suggest that the CXR should not be carefully evaluated** from a radiologist; **The final diagnosis is always responsibility of an experienced radiologist.**
- The model can be also used **to support the redaction of the report from radiologists**. **It can highlight**, superimposed to the original CXR, an "**area of attention**", where the model has found signs of a disease.

Device limitations:

- The model has not been tested in a pediatric setting, in other words not enough tests have been conducted with patient of age less than 10 years old.
- Not enough tests have been conducted with patients of ages greater than 80 years.
- The model has been trained on JPEG images with 512x512 resolution; Images must be therefore pre-processed and scaled to this resolution, before submission to the model. Pre-processing steps are part of the software provided.
- The model has been trained only on images containing "No Findings" or signs of the following diseases: Atelectasis, Cardiomegaly, Consolidation, Edema, Effusion, Emphysema, Fibrosis, Hernia, Infiltration, Mass, Nodule, Pleural Thickening, Pneumonia, Pneumothorax. The presence of signs of other diseases may lower the resulting accuracy.

Critical Impact of Performance:

Within a set of CXRs produced in a hospital we must expect that those containing signs of pneumonia are a minority (in a percentage which, taking into account the characteristics of typically hospitalized patients, can be between 20 and 35%[1]).

For a device, which cannot be 100% accurate, designed to quickly report suspected cases of pneumonia, a **first very important requirement** is to produce the minimum possible number of false negative (FN) cases.
For such cases, the risk is not to give adequate priority to the thorough examination by an expert radiologist, a risk that should be kept to the minimum acceptable.

Minimizing false negatives can be translated into producing a Recall, defined as

$$\textbf{Recall}^{2} = TP/(TP + FN)$$

as close as possible to the maximum value: 1.

But, at the same time, a **second important requirement** is not to produce an excessively high number of false positives (FP), which would clutter up the priority queue and make the device ineffective.

This second requirement can be translated into requiring a **Precision**, defined as

$$\textbf{Prec} = TP/(TP + FP)$$

as close as possible to the value: 1.

So, in summary, it is essential to have a high Recall, but it is also important to have a balanced situation, with Recall and Precision values as close to each other as possible.

In general, the Recall can be increased by suitably lowering the classification threshold and the Precision increased by raising this threshold. So, the two requirements are somewhat conflicting, and a balance must be found.

The metric that best measures the balance between Precision and Recall is the **F1-score**, which is the higher and closer to 1 the more balanced the Precision and Recall values are.

Based on these considerations, in choosing the probability threshold used to discriminate between negative and positive cases (pneumonia), we followed the following procedure:

1. Determined the curve showing the F1-score trend as a function of the threshold
2. Identified the range of threshold values within which the F1-score is maximum

[1] We have taken into account this percentage in the setup of the test set, that is built with a percentage = 25%.
[2] Recall, also known as Sensitivity, measures in this case the percentage of pneumonia cases correctly identified.

3. Chosen, in this interval, the threshold value that maximizes the Recall (minimizes the number of FN)

In the following diagram, we have plotted the Precision, Recall and F1-score as a function of the threshold, for our model:

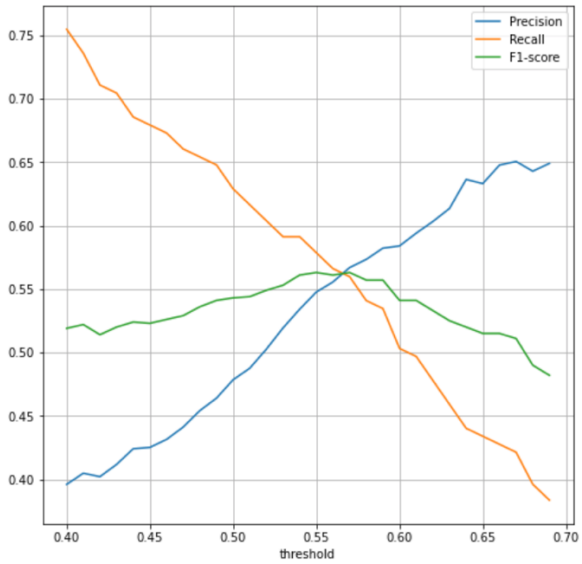

Figure 1: precision, recall and F1-score as function of threshold.

Following this procedure, the most appropriate value for the threshold has resulted as:

Threshold = 0.54[3]

And, for this value we have measured a:

Recall = 0.591
Prec. = 0.534
F1-score = 0.561

A Recall = 0.591 means that our model has showed to be capable to correctly identify on average 6 out of 10 Pneumonia CXR.

The F1-score value, in our understanding, is very good, if compared to similar studies. For example, In Rif. 6[4] it has been reported that, on a similar task, the F1-score for experienced radiologists is around 0.387.

Another important requirement, for the effectiveness of the device as a fast-screening tool, is the speed with which it is able to examine a batch of CXRs and produce a report.

[3] The difference in values for F1-score in the thresholds range [0.54, 0.57] is less than 1% and therefore not statistically significant. Therefore, we have chosen the threshold value (0.54) that satisfy also requirement n.1 (maximize Recall and minimize FN).
[4] References are detailed at the end of the document.

From this point of view, the result obtained is extremely valid. In fact, using a workstation equipped with a **GPU**, the time needed to compute predictions on a batch of 636[5] images is **158 sec**. This time has been evaluated using a server equipped with one GPU NVIDIA Tesla P100.
The time needed to process a single DCM image is about **0.43 sec**., again on a workstation equipped with a GPU.

Therefore, we can affirm that, using this application, we can rapidly assess a single CXR or a large set of CXRs and use the predictions made by the model to place positive CXRs in the high-priority queue, for evaluation from radiologists, with a reasonably low number of FN and FP.

We believe that the application can be useful to avoid that patients with Pneumonia, emerging from CXRs, wait for a long time before the radiological assessment and to give the right priority (for example, based on the predicted probability from the model).

The application can also be used, for the evaluation of single CXRs, on a workstation equipped with a modern CPU (at least 4 core) but without a GPU. In this case, the time for inference on a single CXR is about 0.5 sec.

2. Algorithm Design and Function.

Flowchart:

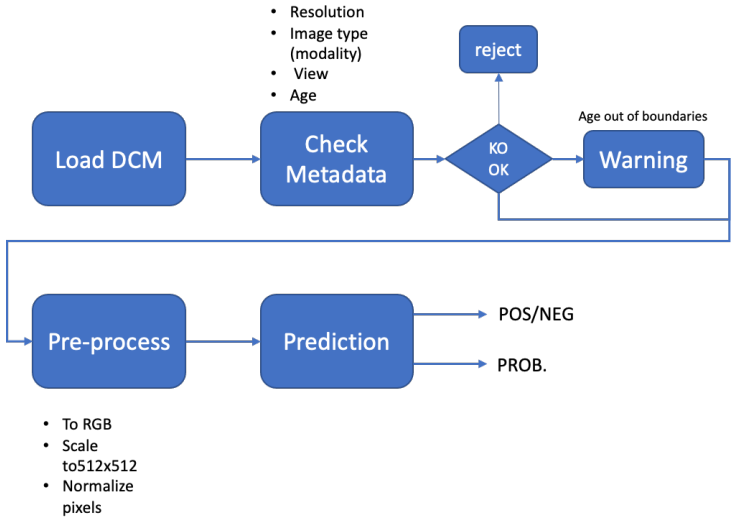

Figure 2: sequence of actions in the algorithm.

DICOM Checking Steps:

The following is the list of the checking steps, before the submission of the image to the CNN:

[5] The size of our test set.

1. The software checks that the image resolution is greater than 512x512 pixels[6]
2. It checks that Image Type (Modality) is **DX** (Digital Radiography)
3. It checks that Body Part is **Chest**
4. It checks that the View Position is **AP** or **PA**
5. It checks that the age is within boundaries (10-80)

If any of the checks 1-4 fail the image is rejected.
If the check n. 5 fails a warning is issued in the report.

Preprocessing steps:

The following is the list of the preprocessing steps, before the submission of the image to the CNN:

- The image is converted from grayscale to RGB (3 channels)
- The image is resized to 512x512 pixels
- The value associated to every pixel is scaled down in the range [0, 1] (divided by 255[7]).

In addition, we have preprocessed the images to build the train-validation and the test set, as described, with more details, in the section 4 of this document (Databases).

CNN Architecture:
In order to achieve accuracy and performance (sensitivity, specificity) in line with the intended use and with the best results in research, a **pre-trained Convolutional Neural Network** (CNN) has been used, applying **transfer-learning**.

A "state-of-the-art" CNN architecture has been chosen: **EfficientNet**.
(see Rif. 2: https://ai.googleblog.com/2019/05/efficientnet-improving-accuracy-and.html).

EfficientNet is a scalable class of networks, proposed by **Google AI Researchers in May 2019**, that provides both high accuracy and efficiency (requires less computational power, both for training and inference). It has been chosen the **EfficientNet B4**, with weights trained on ImageNet.

For the **EfficientNet,** it has been used the implementation available in the PyPI official repository; The implementation's code is publicly available in the GitHub repository: https://github.com/qubvel/efficientnet

The Neural Network's top layers have been replaced by a **classifier composed by three layers**:

- A Global Average Pooling (GAP) Layer
- A Fully Connected, Dense, layer, with 512 units and "ReLU" as activation function
- A Fully Connected, Dense layer, with one computational unit and sigmoid as activation function.

[6] This is needed because the image is resized to 512x512. We don't expect images with smaller resolution.
[7] Images are 8-bit. Therefore, the biggest value for a pixel is 255.

The Neural Network has a total of **18,467,145** trainable parameters.

3. Algorithm Training.

Initial weights for the convolutional part of the network are from the EfficientNet implementation, trained on ImageNet.

In order to **fine-tune** the network on CXR images, it has been chosen a modern approach widely used with pre-trained networks. The approach, for example, is the most widely used in all Kaggle's competitions for Image Classification.

A **custom Learning Rate Scheduler** has been developed and tuned. Initially, a very low learning rate (lr = 1e-5) is applied, to not disrupt the initial weights trained on ImageNet and, at the same time, start to train the final classifier. The learning rate is increased with a linear ramp, then its value is maintained for some epochs. In the final part of the training the learning rate is reduced, with exponential decay.
To maximize the Sensitivity, we have used a very slowly increasing learning rate. Many tests have been done in order to get the right hyper-parameters for the learning rate scheduler.
As a side effect, we have used a high number of epochs.
The following picture show the evolution of the learning rate with the epochs.

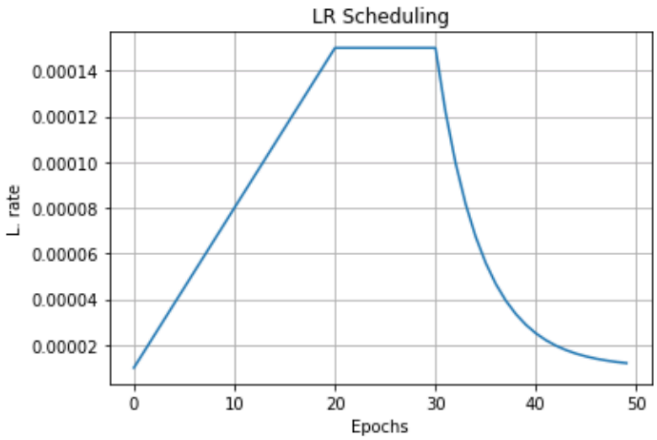

Figure 3: change of the learning rate with epochs.

In order to get a reasonably accurate indication of the performances obtainable on CXR not seen during the training, **K-fold cross-validation has been applied.** For each fold, the training set is split into K parts (K=5 has been used). 80% of images are used for training and 20% for validation. Training is repeated with a different train-validation split and, therefore, **five different models**, one for each fold, **have been trained and saved**.

The final predictions are produced, as average, by the **ensemble** of these five models.

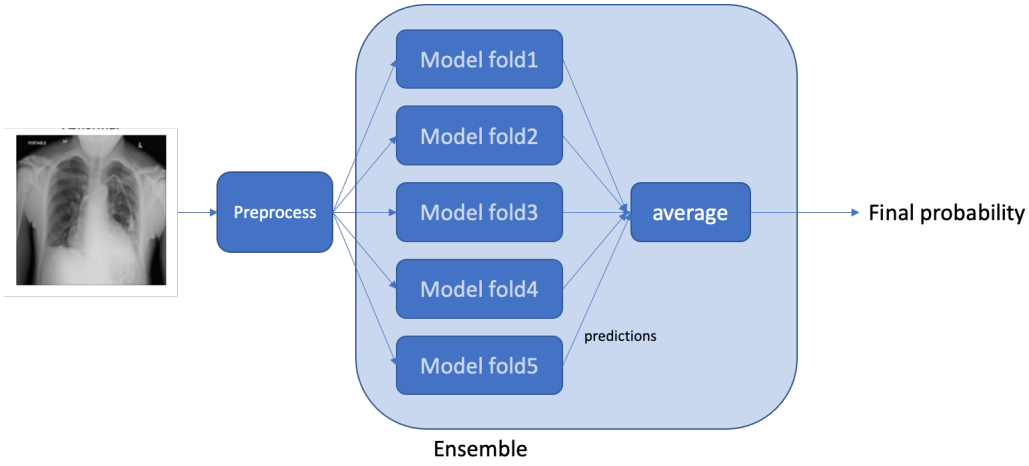

Figure 4: ensemble

The total number of images used for train-validation is **2544**.
During the training, for each fold, **validation loss** is used to monitor the training progress. The **weights producing the lowest validation loss are saved**. In addition, validation accuracy and validation ROC-AUC is monitored during the training.

Parameters used:

- To avoid overfitting, Image Augmentation has been used on the training set. The list of applied transformations is random rotation, shear, zoom, horizontal and vertical shift, flip left-right, random saturation, random contrast, random brightness; A list of the parameters for image augmentation is provided in the additional information section
- No augmentation is applied on images used for validation and test
- Batch size: 16
- Number of epochs: 60
- Optimizer used: Adam
- Learning rate: controlled by a Custom Learning Rate Scheduler (see above)

In order to speed up the training phase and avoid I/O bottlenecks, all the images used for train-validation and for the test have been pre-processed (compressed to JPEG and resized to 512x512) and packed using the **TensorFlow TFRecord** format[8]. This is a binary format, where images and labels are stored sequentially, enabling a very fast loading during the training phase and, therefore, a very high utilization of GPU and TPU.
Image augmentation instead is applied, during the training phase, on-the-fly.

During the hyper-parameter optimization phase, we have used **TPU**[9] (Tensor Processing Units). Using TPU the time needed for training one epoch is about 3 seconds.
The TPU have been provided by the **Kaggle** platform. (see. [Rif. 5])
The best model has been **produced using TPU**.

[8] See: https://www.tensorflow.org/tutorials/load_data/tfrecord
[9] TPUs model is V3-8. Each TPU has 8 processing cores.

The following pictures shows the evolution of key metrics during the training, for one of the folds[10].

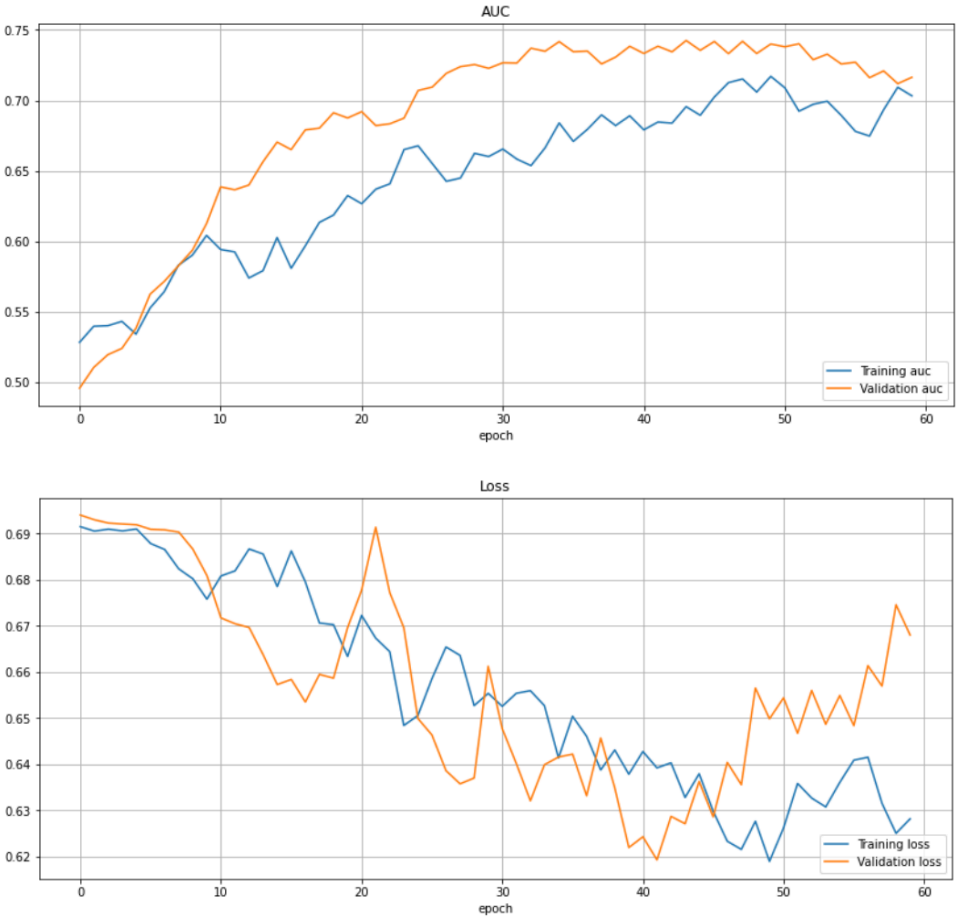

Figure 5

The best value for accuracy[11] and AUC on the validation set have been obtained as the average over all the folds.

AUC = 0.754 +/- 0.006 (std)
ACC = 0.717 +/- 0.043 (std)[12]

A **separate test set** has been used to evaluate AUC, accuracy, precision and recall on samples (CXR) not used during the training phase. The total number of images used for test is **636**.

The probability for the positive case (Pneumonia) is **evaluated as the average of the probabilities produced**, as predictions, by **each of the five models**.

[10] Similar behavior for the other folds.
[11] For the threshold = 0.5. We have reported accuracy, even if it is not the best metric for evaluation in an unbalanced case, since it gives an easy-to-understand indication of the progress of training.
[12] These results have been produced by the inference-tests notebook.

The **Precision, Recall and F1-score** are obviously dependent on the value adopted for the threshold, used to discriminate positive from negative cases.
We have therefore evaluated these **metrics for different threshold values**. The values have been reported in the file: **stat_df.csv**.

Some of the results (close to the maximum F1-score) have been reported in the following table:

Threshold	TP	TN	FP	FN	SENS	SPEC	PREC	F1-score
0.52	96	382	95	63	0.604	0.801	0.503	0.549
0.53	94	390	87	65	0.591	0.818	0.519	0.553
0.54	**94**	**395**	**82**	**65**	**0.591**	**0.828**	**0.534**	**0.561**
0.55	92	401	76	67	0.579	0.841	0.548	0.563
0.56	90	405	72	69	0.566	0.849	0.556	0.561
0.57	89	409	68	70	0.560	0.857	0.567	0.563

Table 1: key performance metrics for different thresholds. Complete list in file: stat_df.csv

The following is the F1-score diagram, obtained on the test set, for different thresholds:

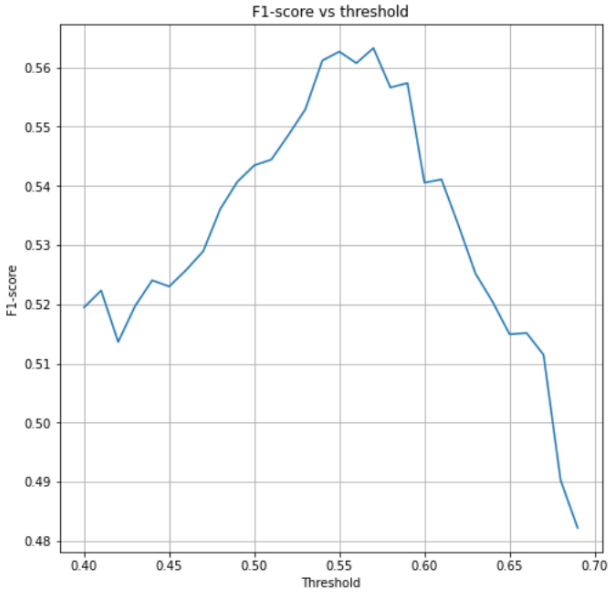

Figure 6: F1-score vs threshold plot.

As explained in the previous section “Critical impact of Performance”, this is the procedure we have followed to identify the best threshold for our model:

1. Determined the curve showing the F1-score trend as a function of the threshold
2. Identified the range of threshold values within which the F1-score is maximum
3. Chosen, in this interval, the threshold value that maximizes the Recall (minimizes the number of FN)

The value selected is:

threshold = 0.54.

For this threshold, we have computed, on the test set, the following metrics:

TP = 94
TN = 395
FP = 82
FN = 65
Recall (Sens) = 0.591
Spec. = 0.828
Prec. = 0.534
F1-score = 0.561

One observation that needs to be made[13] is that the model is today slightly better at correctly identifying negative cases than positive ones. This is, partially, due to the fact that we don't have many pneumonia cases for training and that it is not easy to detect pneumonia from some other comorbidities (see, for example, results from Intensity Profiles analysis).
It will be possible in the future to improve the performance of the model using a higher number of Pneumonia images, during the training.

4. Databases.

Introduction:

All the initial images used to train, validate and test the model have been extracted from the **NIH CXR-14 dataset** (see: https://www.nih.gov/news-events/news-releases/nih-clinical-center-provides-one-largest-publicly-available-chest-x-ray-datasets-scientific-community)

This dataset contains **112120** images. But, as we can see from the following picture, **only 1.3% of these images** have associated the label "**Pneumonia**". T**he dataset is therefore heavily unbalanced** and not effectively usable, "as-is", to train a model for Pneumonia diagnosis.

[13] See the higher value for Specificity, that is defined as P(-|normal): the conditional probability that a normal case is correctly identified by the model.

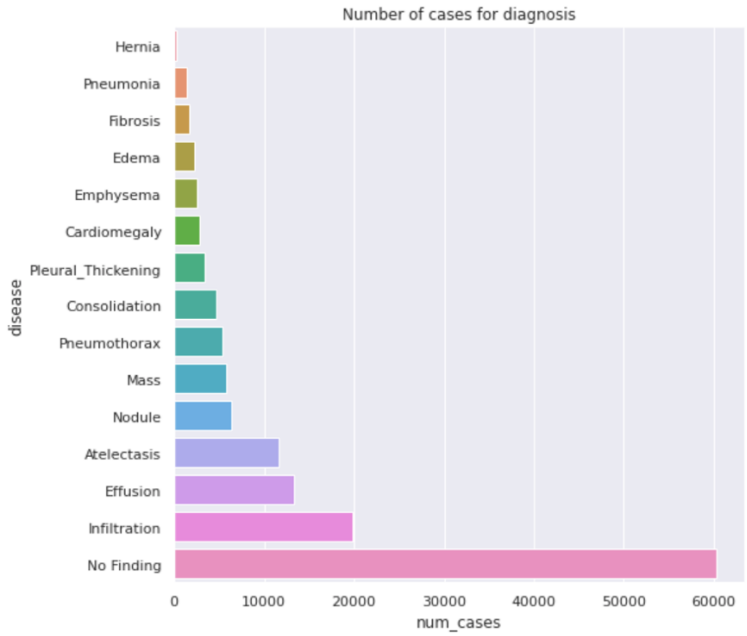

Figure 7

For this reason, we have adopted a set of data preparation **actions to increase the percentage** of "Pneumonia" images in the train-validation and in the test set used, and to get a percentage closer to the one expected in a clinical setting.

Criteria for the preparation steps:

- All the images (1431) with the label Pneumonia have been chosen from the original dataset
- A suitable number of non-pneumonia images, chosen randomly, have been added, in order to satisfy the followings constraints
- A balanced (50% pneumonia, 50% non-pneumonia) dataset for training has been prepared
- A non-balanced test set has been prepared with 25%[14] of images with the label pneumonia
- All the images have been split between train and test set following the rule: 80%-20%
- There is no overlapping between train and test set: no patient has images in both the sets (as seen by the Patient ID)

Putting in place all these criteria[15], this is the resulting composition for the train and the test datasets:

Dataset	n. pneumonia	n. non-pneumonia	Total
Train	1272	1272	2544
Test	477	159	636
Overall			3180

[14] This is close to the proportion we could expect in a clinical setting, where images from patients are been analyzed.

[15] A linear set of equations can be built using these criteria. It has only one solution, whose values are reported in the table n.2.

Table 2: composition for train and test sets.

In the train and test datasets all the images have been:

- Resized to the resolution 512x512
- Compressed in JPEG format (to reduce size on disk)
- Packed in TFRecord format files

Description of the training dataset:

The total number of images in the training-validation dataset is **2544,** belonging to **1896 patients**.
The dataset is **balanced**: the number of positive cases is 50% of the total.

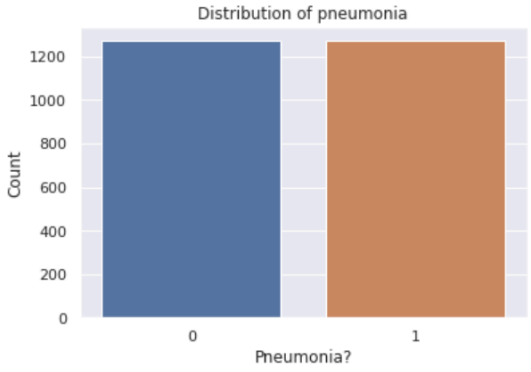

Figure 8: pneumonia in the training-validation set.

The following pictures describe the **distribution of the ages** of the patients in the training set.

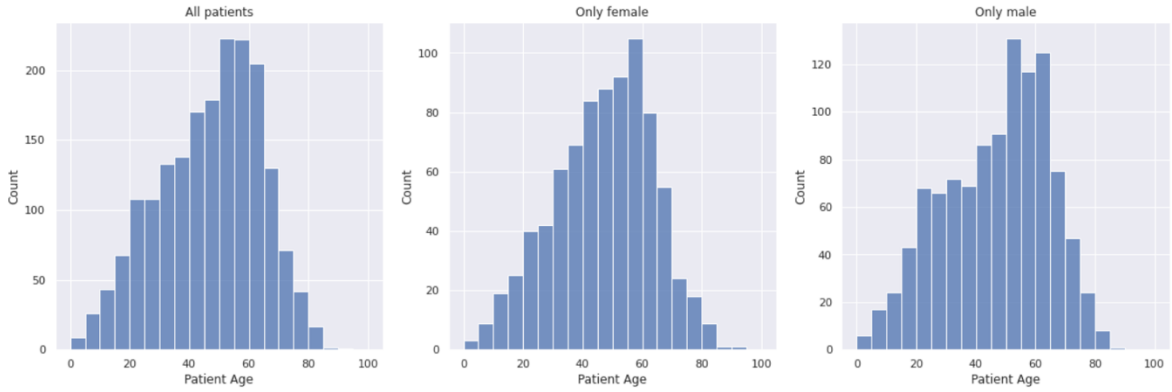

Figure 9

This is the distribution of gender:

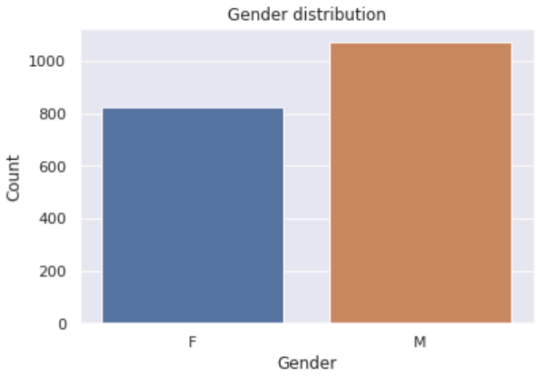

Figure 10: distribution of gender in train set.

We have used K-fold cross validation, as described in the next section.

Description of the validation dataset:

We have applied **K-fold cross validation**, with K=5. For each fold, the train-validation dataset is split into five parts. Splitting is done at the file level.
One part is used for validation and the others for training. In this way, results are quite independent of the splitting between train and validation images.

Figure 11

Description of the test set:

The total number of images contained in the test set is: **636**.

From the following picture we can verify that **the test set used is not balanced**. This is useful to get a more accurate idea of the performances that can be achieved in a real scenario[16].

[16] Where we can expect a percentage of pneumonia cases between 20% and 35%.

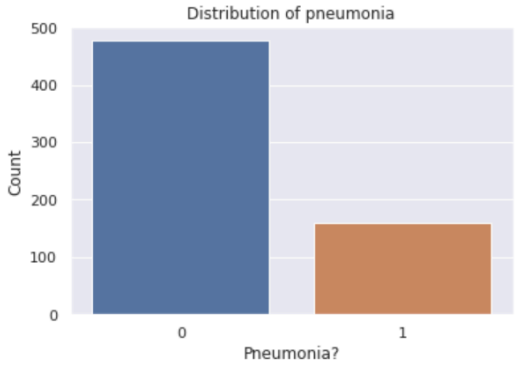

Figure 12: pneumonia in the test set.

The following pictures describe the distribution of ages in the test set.

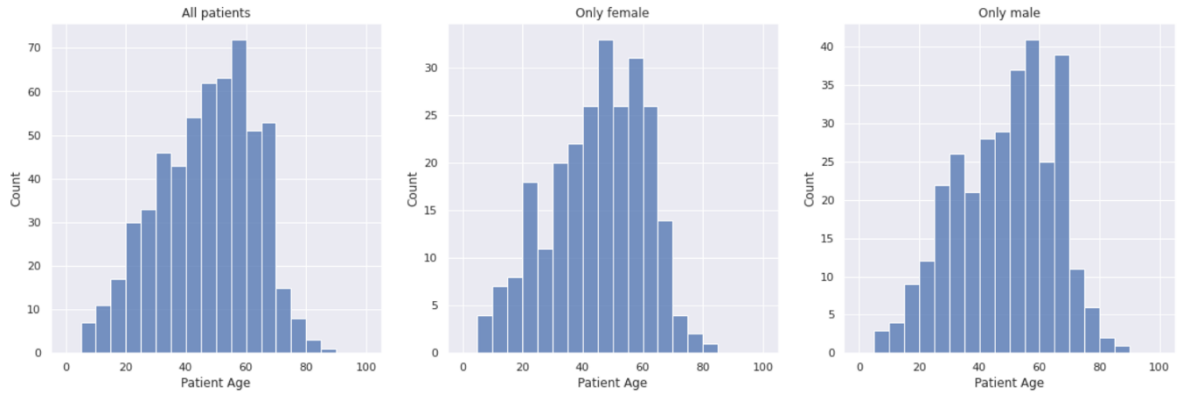

Figure 13

In the preceding picture, the only noteworthy detail is that there are not enough patients over the age of 80. For this reason, we have indicated not to use the model on images of patients over the age of 80.

This is the distribution of gender:

Figure 14: distribution of gender in the test set.

Comparison train – test set.

From the pictures, we can confirm that the distribution of gender and ages is almost the same in the two datasets, without significant differences.

5. Ground Truth.

In our test set the labels (0 = negative, 1 = positive-pneumonia) have been defined through NLP extraction from radiologists' reports and checks from experienced radiologists, as originally from NIH-CXR-14 dataset.

However, it should be noted that pneumonia diagnosis from CXR is not always easy and, in practice, it can happen that two radiologists don't agree. In addition, it is well known that NIH-CXR-14 labels contain several mistakes. See, for example, the analysis contained in Rif. 4.

Therefore, we are convinced that to successfully and reliably develop such a study, the **consensus between a team of experienced radiologists** should be used as **ground truth**.
This is the approach that we will be using for the FDA Validation Plan.

6. FDA Validation Plan.

The FDA Validation Phase will be conducted with the support of the XXX Hospital, which will act as the Clinical Partner.

We will provide the needed installation of the software and all the training and needed support, to enable proper use of the equipment, in accordance with the defined Statement of Intended Use and Indications for Use.
The provided software will be integrated with the PACS system of the Hospital.

XXX Hospital Medical Team will select a set of at least **1000** patients.

The patients will be randomly chosen between those for which a Chest X-Ray is requested and produced. The body part in the radiography must be Chest and the view position must be AP or PA.
For each patient only one CXR will be provided.

The FDA validation set must contain a suitable number of "Pneumonia" CXR. The percentage must be not lower than 25%, following the need to evaluate the model over an enough high number of positive cases.

The FDA validation set will have a distribution of ages as close as possible with those shown by the fig. n. 12. Only patients with age between 10 and 80 years old will be included.

The validation set will contain images from patients as close as possible equally distributed between females and males (50-50%).

The "labels" for this validation set will be defined by a team of **at least three experienced radiologists**, who will review the existing radiologists' reports, along with the CXRs. The resulting labels (negative, positive) will come from the consensus of the radiologist team.
The radiologists will use DICOM images.

The predictions from the overall model will be produced using the **ensemble of five models** and **averaging the probabilities from each one**. The software produces all the needed computations.

The model will use**,** to discriminate a negative from a positive case, **a classification threshold equal to 0.54**. In other words, if the average probability is greater than 0.54 the CXR will be classified as pneumonia.

The model will be evaluated on the FDA validation set **only once**.

The expected results must be in accordance with the results reported in the section: "**Critical impact of Performances**".

To be more precise, the expected **Recall**, that defines the probability that a pneumonia CXR is correctly detected (P(+|disease), must be:

Recall >= 0.591

And, at the same time, the F1-score must be:

F1-score >= 0.561.

In addition, we expect to have an overall accuracy not lower than 0.73[17].

7. Additional information.

Area of interest:

As pointed out in the Indications for Use section, in case of a positive (pneumonia CXR) prediction, the application can also provide a graphical explanation, providing an image where the area most relevant for the positive prediction is highlighted.
The image is produced using the **Grad-CAM** technique.

This is one example:

[17] Computed from the data in table n.1, for the threshold = 0.54.

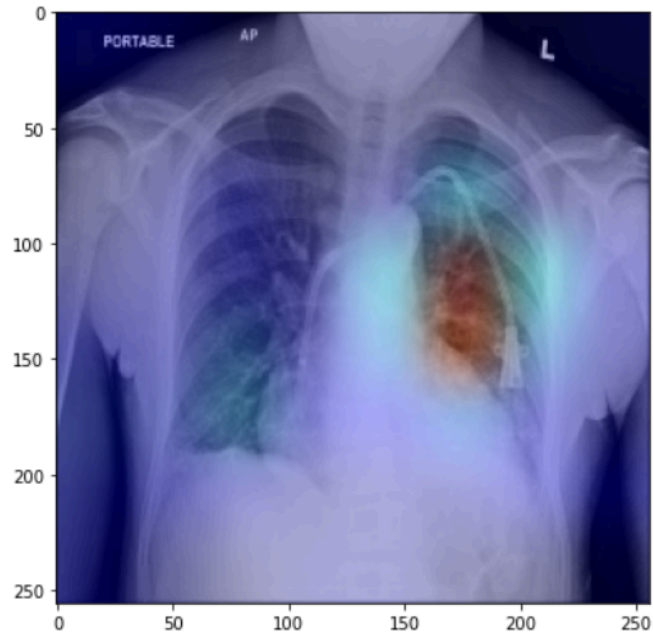

Figure 15: positive CXR with highlighted the area most relevant for the prediction.

For more details regarding Grad-CAM see [Rif. 3].

Image augmentation: parameters.

This is the list of transformations applied to images in the training set for augmentation:

- Random rotation
- Shear
- Horizontal zoom
- Vertical zoom
- Random saturation
- Random contrast
- Random brightness

These are the parameters used in the code used for the training of the CNN:

- ROT_ = 10.0: maximum angle for random rotation
- SHR_ = 2.0: maximum shear angle
- HZOOM_ = 4.0: parameter for horizontal zoom (fraction of total height)
- WZOOM_ = 4.0: vertical zoom (fraction of total width)
- HSHIFT_ = 4.0: horizontal shift (fraction)
- WSHIFT_ = 4.0: vertical shift (fraction)

These parameters have the same meaning of the corresponding's in the class:

tf.keras.preprocessing.image.ImageDataGenerator.

For more information, see, for example:

- https://keras.io/api/preprocessing/image/
- https://www.kaggle.com/cdeotte/rotation-augmentation-gpu-tpu-0-96, section "Data Augmentation".

References:

[Rif. 1] Automated abnormality classification of chest radiographs using deep convolutional neural network, Yu-Xing Tang et al., Nature Digital Medicine, May 2020, https://www.nature.com/articles/s41746-020-0273-z

[Rif. 2] **EfficientNet:** Improving Accuracy and Efficiency through AutoML and Model Scaling, Google AI Blog, May 2019, https://ai.googleblog.com/2019/05/efficientnet-improving-accuracy-and.html

[Rif. 3] **Grad-CAM**: Visual Explanations from Deep Networks via Gradient-based Localization, R.R. Selvaraju et al., arXiv.org, Dec. 2019, https://arxiv.org/abs/1610.02391

[Rif. 4] Exploring the **ChestXRay14 dataset: problems**, L. Oakden-Rayner, Dec. 2017, https://lukeoakdenrayner.wordpress.com/2017/12/18/the-chestxray14-dataset-problems/

[Rif. 5] Our best model's training on TPU, Kaggle, https://www.kaggle.com/luigisaetta/cxr-pneumonia-notebook3

[Rif.6] CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning, Pranav Rajpurkar et al. section n. 4, https://arxiv.org/pdf/1711.05225.pdf