Project Definition

Project Overview

Student provides a high-level overview of the project in layman's terms. Background information such as the problem domain, the project origin, and related data sets or input data is given.

As a biomedical Engineer, I decided to work on a dataset related to my field of study. Which is treating data of clinical nature. I consulted Kaggle to find a chest x-ray dataset. Based on the type of the data and being motivated to artificial intelligence based diagnosis, I decided to exploit this dataset to build a solution that can achieve a classification between pneumonia affected or healthy chest x-rays. Before this, let's gain some basic understanding about this pathology to help efficiently design and understand the adding value for this solution.

The following explanation is inspired from: www.mayoclinic.org

Pneumonia is a generally a bacterial or viral infection that affect the lung alveoli. Symptoms include fever, productive or dry cough, difficulty of breathing, chest pain....

Different risk factors can cause pneumonia such as chronic diseases like asthma and heart diseases, smoking, diabetes and weakened immune system.

Pneumonia can be diagnosed (based on www.nhlbi.nih.gov) by a review of the medical history perform blood tests; require chest X-ray and pulse oximetry or chest computed tomography CT scan. The diagnosis can conclude if the patient has pneumonia or not. In case of disease, a further classification can also be concluded depending on where the pneumonia was acquired such as community, hospital acquired, or healthcare-associated pneumonia

Problem Statement

The problem which needs to be solved is clearly defined. A strategy for solving the problem, including discussion of the expected solution, has been made.

As we saw in the project overview, Pneumonia diagnosis is based on different parameters. However, in emergency cases, especially for bacterial type of pneumonia where taking antibodies rapidly is crucial, performing blood analysis can take too long before the results are accessible for the physician. Hence, building a diagnosis initial decision based only on X-ray chest images (a rapidly accessible information) can encourage the doctor to decide whether the patient should start the therapy or not. The aim of this project is thus to build a classifier

that can decide if a patient has a pneumonia by providing only its chest X-ray in order to help the physician to decide the further handling of the patient.

Metrics

Metrics used to measure the performance of a model or result are clearly defined. Metrics are justified based on the characteristics of the problem.

As a classification problem, we will use accuracy precision and recall as metrics to measure performance of the model.

Let TP, TN, FP and FN be true negative, true negative, false positive and false negative respectively we recall that:

$$Accuracy = (TP + TN)/(TP + FP + TN + FN) = \frac{Number\ of\ correct\ predictions}{total\ number\ of\ predictions}$$

$$Precision = TP/(TP + FP)$$

$$Recall = TP/(TP + FN)$$

Analysis

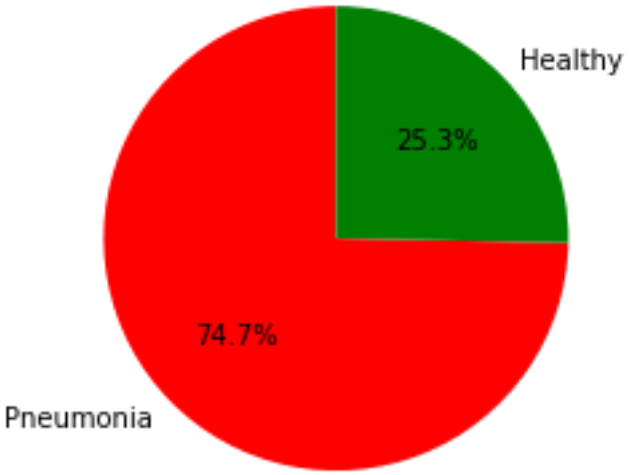
Data Exploration

If a dataset is present, features and calculated statistics relevant to the problem have been reported and discussed, along with a sampling of the data. In lieu of a dataset, a thorough description of the input space or input data has been made. Abnormalities or characteristics of the data or input that need to be addressed have been identified.

After downloading the dataset from the provided directory, we find that the main folder is composed three sub folders (train, validation and test) of chest x-ray images respectively. Each folder contain two subfolders named: NORMAL and PNEUMONIA. The class of each data point (image) is deduced from its location. At a first glance, we can notice that images has different spatial sizes. In addition, some are rgb and others are gray level images.

The number of validation images is much less, than the number of train images (16 vs 5216 data point). To avoid this first imbalance, we stack the validation data to the training data. We shuffled them and split it according to the 0.2 ration. Now we have 4185 training data point vs 1047-validation data point. This step was crucial, as validation performance is the deciding for the training efficiency of the model.
We have also noticed another type of imbalance that is classes' imbalance. In

the training data, we have 3126 Pneumonia labeled image vs 1059 Normal labeled image.



Data augmentation in this case (rotations, modifying the pixels intensities, flips…) however, those transformations are not allowed, as chest x-ray images are not expected to be rotated or flipped. Thus, such data would not exist in the reality. Overcoming this class imbalance by other means is thus obligatory to guarantee a fair training process of the model. The figure below is a visual illustration of the mentioned classes' imbalance:

Exploratory Visualization

A visualization has been provided that summarizes or extracts a relevant characteristic or feature about the dataset or input data with thorough discussion. Visual cues are clearly defined.

In the following, a random set of Pneumonia labeled and Normal labeled images:

The image is almost centered x-ray visualization of patient chest on a dark background. One can easily notice that the lung is a bright region merged in the middle of the thorax and located in the middle region of the frame. It is easy to notice that in generally the pneumonia affected lungs present higher volume than the healthy one. In some cases however, it is difficult to differentiate, the healthy image to the ill one. We can consider for example the last Normal image the first Pneumonia one. As we mentioned in the project overview the diagnostic is based on different parameters. As even for experts, only chest X-ray based decisions are difficult to take. Thus building such a model can considerably help the physician solidifying their diagnosis.

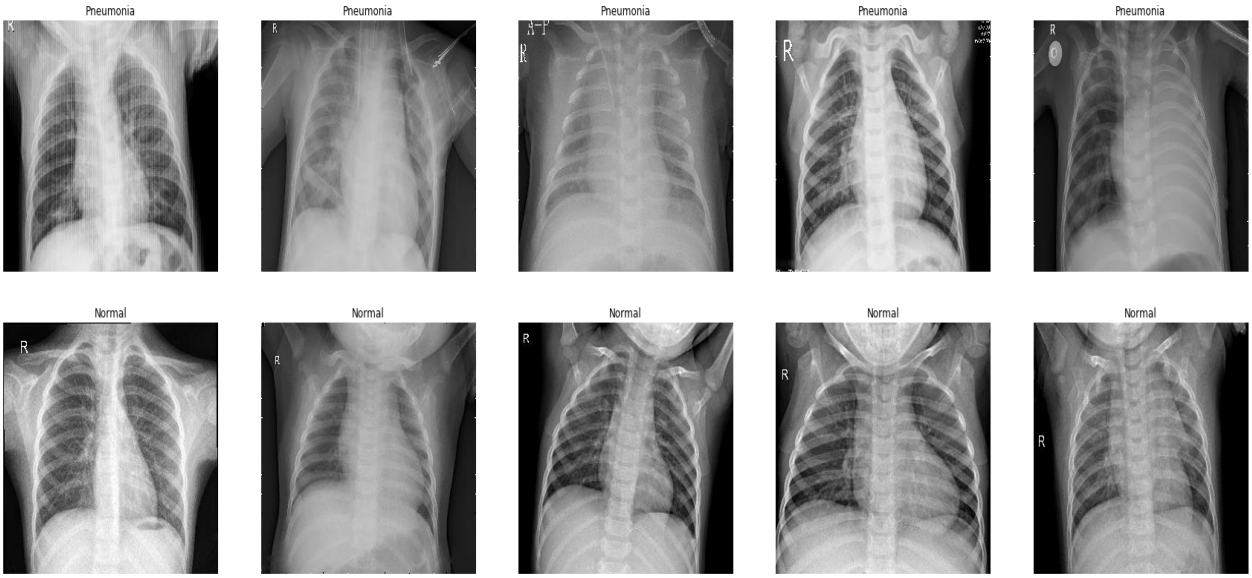


Fig 1 a Set of Pneumonia and Healthy samples

Algorithms and Techniques

Algorithms and techniques used in the project are thoroughly discussed and properly justified based on the characteristics of the problem.

CNN are the most adopted deep learning model for computer vision (because it save the 2-D nature of the image and consider the neighborhood of the pixels). We adopted the vgg16 CNN model as it present a fair depth for our task. Since our problem is, a binary classification one we stack fully connected layers FC to the CNN that has a binary output. We selected "ReLu" as nonlinear activation function, vgg16 keras implementation has a default padding value equal 'same', this is to preserve the spatial dimensions as the input when stride is set to one). Another reason for adopting this CNN model is that it uses the depth wise separable layers to optimize the number of learning weights, alleviate consequently the learning task and reduce the needed resources.

The following table is summary of the model layers and the overall architecture adapted from the original architecture presented in [1]:

Table 1 Model Layers and Architecture

Layer type	output channel	prevalence	Kernel size
Conv2D	64	2	(3,3) & (3,3)
MaxPooling2D	-	1	-
SeparableConv2D	128	2	(3,3) & (3,3)
MaxPooling2D	-	1	-
SeparableConv2D BatchNormalization	256	2	(3,3) & (3,3)
SeparableConv2D	256	1	(3,3)
MaxPooling2D	-	1	-
SeparableConv2D Batchnormalization	512	2	(3,3),(3,3)
SeparableConv2D	512	1	(3,3),(3,3)
MaxPooling2D	-	1	-
Flatten	100352	1	
Dense	1024	1	
Dropout	1024	1	0.7
Dense	512	1	
Dropout	512	1	0.5
Dense	2	1	

Benchmark

Student clearly defines a benchmark result or threshold for comparing performances of solutions obtained.

We found on Kaggel multiple notebooks that use the mentioned dataset for the defined task. Accuracy was the criteria used by the platform to rank contributions. In our wok we will focus on the precision and recall metrics since we believe that they can better express model reliability for two reasons, namely class imbalance and avoiding FN. In fact, FN is the parameter most to decrease for this classification. This last parameter gives rise to wrong assumption to the physician and cause ignoring real patients cases to be further treated. FP in the other way, is not a big concern in this case as it only would cost a further handling for a healthy individual. We commit ourselves then to achieve a respectable recall amount to guarantee a reliable performance of this solution.

To conclude, if we achieve the claimed 94% accuracy PLUS a high recall then we can consider this project as value adding one. In the following section, we will go into deeper details about data processing and model learning operations.

Methodology

Data Preprocessing

All preprocessing steps have been clearly documented. Abnormalities or characteristics of the data or input that needed to be addressed have been corrected. If no data preprocessing is necessary, it has been clearly justified.

For data preprocessing we performed a number of operation to adapt data to be a reliable input for the suggested model. First, we decide to resize all images to have a 224 by 224-spatial size, as it is a recommended size in the literature. Second, we decide to affect initially the label 0 to healthy scans and label 1 to sick ones. Finally, in order to counter act the mentioned classes imbalance without using data augmentation technics by changing the label values following the calculations above:

We start by calculation the initial bias between classes

$$initial\_bias = np.log(count\_pneumonia/count\_normal)$$
$$initial\_bias = 1.082$$

The allocated weights to the 0 and 1 classes are determined as follow.

$$weight\_for\_0 = (1/count\_normal) * (count\_train)/2.0$$

$$weight\_for\_1 = (1/count\_pneumonia) * (count\_train)/2.0$$
$$Weight\ for\ class\ 0: 1.97$$
$$Weight\ for\ class\ 1: 0.67$$

We assigned labels for every image based on the directory (as explained in data exploration section)

Third, we decided that all images has to have three channels (as the number of RGB images is more than the gray level images). Which means gray level images acquisition was performed in a way that the output images has three channels. This can be achieved by different means. We used for this operation the *tensorflow decode_jpeg* function.

Implementation

The process for which metrics, algorithms, and techniques were implemented with the given datasets or input data has been thoroughly documented. Complications that occurred during the coding process are discussed.

The implementation part split into two main parts such: model learning and learning evaluation.

The learning process starts by building a data generator (*buildxy function)* for building the pre-processed training and validation data. This data was then saved in an "upload" directory. Data will be called consecutively during the model fitting. Validation data however, will be called all at once. This mean that model performance estimation will not be executed on a sequential manner with the validation data but at once.

Model design was built by using loading the keras layers of the vgg16 model. Weights for vgg16 trained on ImageNet dataset [2] will be only kept for layers 1, 2, 4 and 5 as a transfer learning procedure. We will back to our transfer leaning choices in more details in the next section. The learning process optimizer was choose for ADAM optimization algorithm. Learning rate lr is set 10^{-4} a decay rate of $10^{-5}$is applied every epoch. The mentioned accuracy metrics were also set and fed to the *model.compile()* method. Obviously, the loss function was set to *binary-crossentropy* as the task is binary classification one. We also kept the default *softmax* activation function. We choose a batch size of 16 for the training data and we decided to train the model for 20 epochs.
We choose also to save the model state each epoch as a checkpoint to enhance the fine-tuning operation by selecting best model. This practice also is very important to resume learning after training interruption.

Refinement

The process of improving upon the algorithms and techniques used is clearly documented. Both the initial and final solutions are reported, along with intermediate solutions, if necessary.

In order to improve the learning speed and performance we adopted a transfer learning procedure. We froze the shallow weights layers of the the well-known vgg16 [1] model by the initial weight learned from the ImageNet dataset. This procedure is based on the assumption that our dataset has a relative big size and belongs to a different context comparing to the ImagNet dataset. The last mentioned two factors will dictate the way in which we perform transfer learning procedure. The not small size of the dataset will allow to fine-tune the initialized layers from VGG16 model (but with small leaning rate) and the big difference between datasets contexts will restrict as to transfer learning only to the first layers of the model as the last ones will present features that are more linked to the training dataset. A second action for refinement is that we set the *restore_best_weights* to true in the defined keras callback. This will allow that we keep the best version of the model that was trained during all epochs.

Results

Model Evaluation and Validation

The final model's qualities—such as parameters—are evaluated in detail. Some type of analysis is used to validate the robustness of the model's solution.

In the following, we visualize our model performance. The figure below visualize the model precision, accuracy recall and loss of the train data vs the validation data. Please not that this figure is not the same as the one in the submitted note-book because the last one does not present validation (validation data lost)

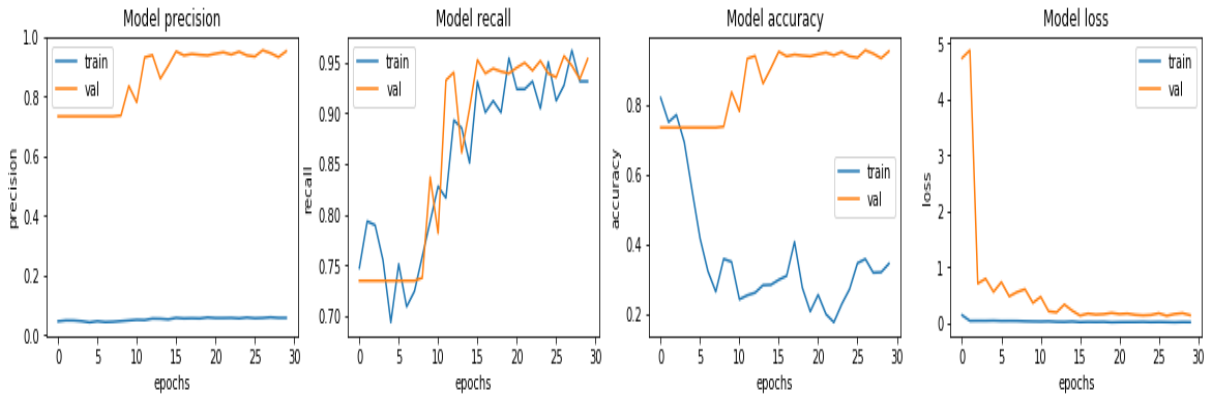
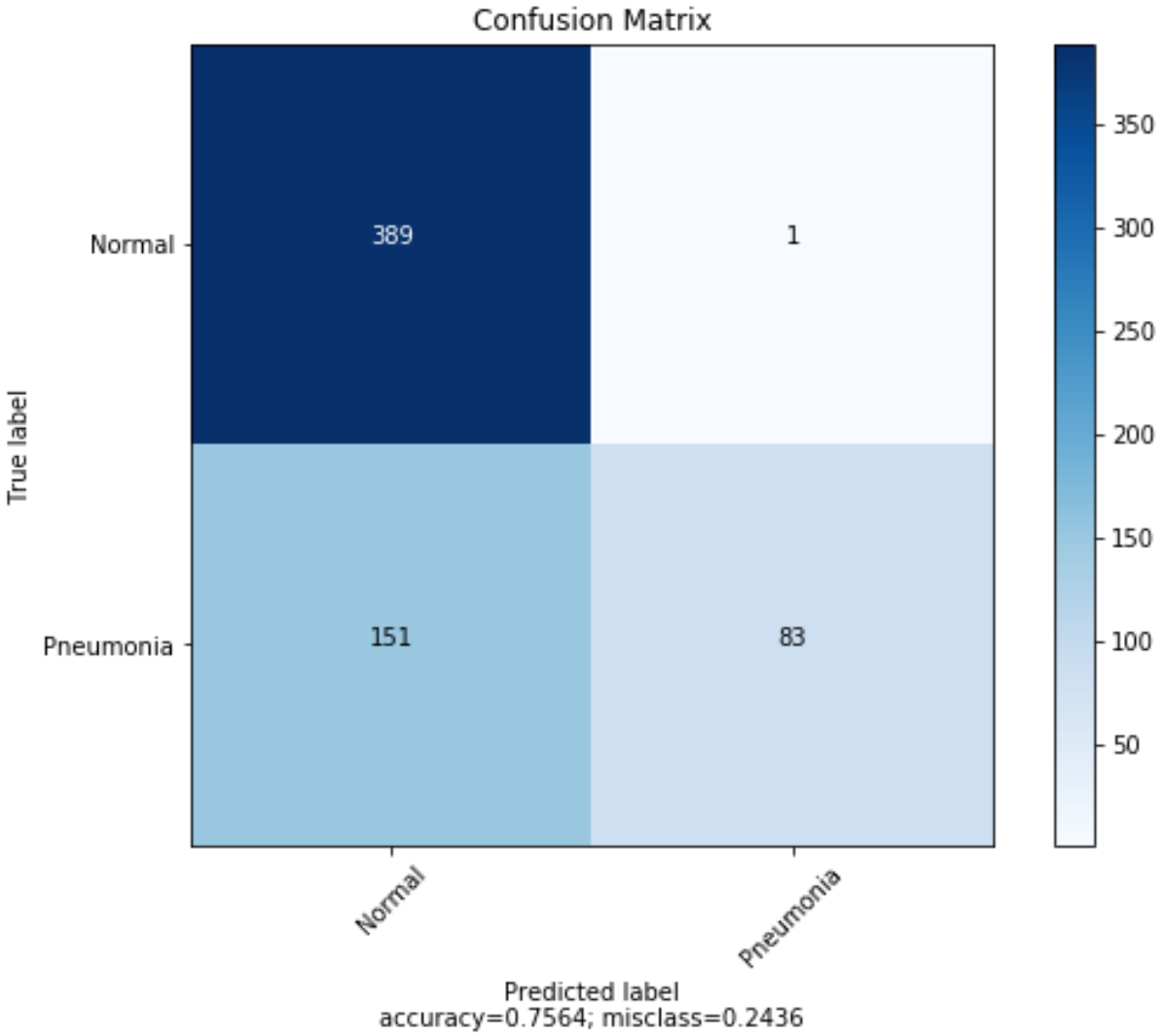


Fig 2. model performance on validation Data

The benchmark claimed accuracy on the validation data was achieved. The added value of this work that we proved a high recall and precision of the model that better justify the classification reliability.

The poor performance in the training data can be justified by the two dropout layers mentioned in table 1 (this push the model generalization ability by shutting down a number of neurons randomly) of the model. Such a layers put the training data in more challenging conditions than the validation one since the dropout is deactivated when running the validation data.

In the following, we present the confusion matrix calculated on the test data. We find an accuracy, precision and recall of all 75.7%. Even with a less accuracy level comparing to the benchmark, we can more rely on our model predictions based on 75.7% of recall. This model was trained for only 20 epochs and almost 20% of the training dataset was allocated to the validation dataset. We could compensate this training data decrease by further training this model. A better performance should then be expected.



Justification

The final results are compared to the benchmark result or threshold with some type of statistical analysis. Justification is made as to whether the final model and solution is significant enough to have adequately solved the problem.

In the mentioned work at the benchmark section, the author choose to set only the accuracy as a metric to optimize. In our work, we also set precision and recall as an additional criteria to optimize. This is very substantial for this specific dataset. The class imbalance that we clearly presented in the data exploration section can give a wrong impression by only looking to the accuracy. A classifier that only output “1” to each training sample would achieve 75% accuracy. However, the more important question we want to answer is “what proportion of actual positive samples was identified correctly?” Classifying a healthy patient as sick will not cause a big issue, as this would only encourage the physician to perform more test. In the other hand, discouraging the doctor to perform further analysis for a sick person is a real

issue. Answering the mentioned question is equivalent to monitor the recall parameter.

References:

[1] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556* (2014).

[2] http://www.image-net.org/