Lung Diseases Classification using X-ray images and machine learning

|  |  |
| --- | --- |
| 1st Ahmad Saigol  *M.Sc. Mechatronics*  *Technische Universität Hamburg*  Hamburg, Germany  AhmadSaigol1@gmail.com | 2nd Joel Romero  *M.Sc. Mechatronics*  *Technische Universität Hamburg*  Hamburg, Germany joel.romero.munoz@tuhh.de |

*Abstract*—COVID-19 continues to be a global health affair. Early detection through CRX Images is an important countermeasure. For this study, feature extraction together with traditional machine learning methods, as well as deep learning methods, have been used. Feature extractors Haralick and Zernike moments, together with Support Vector Machine as classifier, obtained the highest balanced accuracy within an ML approach: 88.03% for binary and 82.01% for multiclass classification. The pretrained neuronal network MobileNet V2, together with Online Augmentation, provided the highest balanced accuracy of 90.67% for binary and 87.59% for multiclass classification.

# I. INTRODUCTION

## A. Background and motivation

On March 2020, the COVID-19 outbreak was declared a pandemic. Ever since, vaccines have been developed, but they have an efficacy of 95% only in terms of preventive action rather than a cure per se. Therefore, early diagnosis of COVID-19 is still important, as it can be traced to avoid further spread. To accomplish this, computed tomography (CT) and chest X-ray (CXR) images are used by healthcare professionals. Currently, they are focusing efforts on early detection of coronavirus: only in the first 80 days of 2020, about 1245 academic articles were written. Deep Learning (DL) is an State of the Art method for COVID-19 detection using CXR and CT Images. [1]

## B. Problem overview

This study aims to perform COVID-19 detection and differentiate it from other respiratory conditions, by contrasting or combining feature extraction with traditional machine learning methods, and by using deep learning. Deep learning offers the advantage of feature selection not been hand-picked, in contrast to traditional machine learning methods. A specific CXR database was provided. Two types of classification were required: Binary and Multiclass. Binary Classification would determine whether a patient had Covid or not. Multiclass Classification would diagnose 4 different conditions: Healthy (Normal) lungs, Covid-19, Pneumonia and Lung Opacity.

## C. Objectives

This study was divided into two phases. During Phase 1, the use of feature extractors and supervised machine learning models was demanded for Classification. During Phase 2, the use of Deep Learning models was required. In the end, the major objectives are:

* Evaluate how preprocessing affects the performance of the classification models.
* Evaluate the effect of feature extractors hyperparameters on overall model performance.
* Determine an appropriate Classifier for ML models.
* Analyze the impact of Augmentation in DL models.
* Contrast the performance of ML models against DL models, plus the advantages and disadvantages of using each of them.

# II. RELATED WORKS

Several studies are being conducted to find artificial intelligence solutions for respiratory diseases detection, and specially for COVID-19 due to the current worldwide health status.

Emre A. conducted a research using ML, which consisted of applying segmentation to CRX images, to extract only the area of interest by masking the images with a circle that shares the same center as the chest. Posteriorly, the images were resized to 15x15 pixels and taken to the RGB color space. Each one of the 255 pixels’ intensity value were a used as a feature vector for 5 different Machine Learning algorithms: studies, 5 different Machine Learning Algorithms (MLAs): Multi Class Support Vector Machine (MC-SVM), k Nearest Neighbor (k-NN), Decision Tree (DT), Multinominal Logistic Regression (MLR) and Naive Bayes (NB). SVM outperformed with 93% on the test success [2]. Juliana Gomes et al. developed an intelligent tool for COVID detection without segmentation. In this case, Haralick and Zernike moments were instead used as feature extractors. Multilayer Perceptron was also tested as a classifier. In the end, the classifier SVM outperformed with a balanced accuracy of 89.78% [3].

Geeta Rani et al. performed an extensive study with a DL approach. Preprocessing was a step were researchers put a lot of emphasis. Firstly, the dataset was analyzed by radiologists to separate bad quality CRX images, which are produced by a lack of respiratory capacity in sick patients and can undermine the performance of the model. Then, bony structures such as neck, arms, ribs, collar bones, and solid tissues outside the lung region were removed as they interfere in the feature extraction process. To do this, a bone shadow suppression model and a lung segmentation model were applied, and as a result, only the lung region was extracted from the CRX. Finally, the extracted lung areas were used as input for an EXP-Net model. They determined that lung segmentation increased the DL classifiers’ overall accuracy by an average of 2.7% and bone suppression increased it by 4%. Also, EXP-Net performed better than different classification models such as ResNet50 or ResNET101, and achieved an accuracy of 94.07% [4]. Dongguan Li et al. developed a DL high accuracy platform to detect Covid-19 from CRX images. The core approach of the study was to generate a voting algorithm that combined 17 pretrained CNNs (including ResNet18, RestNet50, MobileNetv2), to overcome image variations. Five different binary classifiers were used to assign the images into 4 categories: Healthy, Bacteria, Covid-19, and other virus. An accuracy of 99.62% was obtained when averaging the performance of all the classifiers, i.e., the final accuracy of the proposed model [5].

# II. Materials and Methods

## A. Dataset Description

To perform this study a dataset of Chest X-Ray images inside different folders was provided, which included:

* A folder with 16930 X-Ray Images and its labels were provided to train the Machine Learning models.
* A folder with 4235 images was also provided to test the accuracy of the model.
* A folder with noisy dataset 4235 was provided to test the DL models developed in Phase 2 [6].

## B. Data Splitting

The images of the training set were split into training and validation subsets. Random generation of subsets is not always the most appropriate method, as the distribution of the target variables can differ between datasets. Moreover, in the case of class imbalance within the training dataset, where instances of the target variables are not evenly distributed, intended model’s performance can be reduced even more. This was the case for the provided dataset, as shown in Table 1, where the number of instances for the class “Normal” exceeds by four times the number for “COVID”. Hence, the Stratified Sampling technique was applied during data splitting. It consists of forcing the distribution of instances of the target variable(s) to be the same among different subsets [7]. The raw data was then split into 70% for the training subset and 30% for the validation subset.

Table 1 Classes distribution within raw dataset.

|  |  |
| --- | --- |
| **Class** | **Instances** |
| Normal | 8153 |
| COVID | 2892 |
| Pneumonia | 1076 |
| Lung Opacity | 4809 |

## C. Data Preprocessing

To effectively process the information, different operations where performed to the images within the training subset. These operations mentioned bellow were tested in different combinations.

Normalization: It was applied so values obtained from feature extractors can be comparable. Min-max normalization method takes the pixel values of the image to a range from 0 to 1, which is given by (1):

where represent the image pixel’s intensity value, and the maximum and minimum intensity value of the image [8].

Resizing: Images of the same dataset were converted to a determined same size, so ML algorithms could handle them in a consistent manner. Additionally, pretrained Neuronal Networks (NN) require a fixed input size which are optimal, so images had to be adapted to the appropriate size depending on the NN to be trained.

Bilateral Filter: It performs a pixelwise operation that uses the neighborhood pixels to determine the spatial and intensity distance between them. It was used as it reduces any existent noise within the image, while preserving the texture of the image, which is decisive for the problem being tackled in this research [9].

Augmentation: The noisy data test provided was mainly composed of rotated X-Ray images (90° and 180°), which represented an additional challenge to solve for the DL models. To overcome this, Augmentation was applied to the training dataset. It consists of applying different operations to existent images and storing the results. Applied operations included: translation, rotation, horizontal flip, vertical flip, and cropping. It should be noted that Augmentation is usually used when there is an insufficient volume of data to train the model [10], which was the case for the CNNs’ training of the present study, as it provided noisy images to the training dataset. Online and Offline Augmentation were applied to train the Deep Learning models. Online consisted of applying Augmentation to the input of the model, i.e., all the images of the training subset, while training it, by randomly applying the operations previously mentioned individually or as a combination of them. Offline consisted of generating additional images by augmenting the training dataset, preserving the tags from the original images, and adding the results to indicated dataset prior to starting to train the model.

## D. Feature extraction

During Phase 1, Machine Learning models required an input with information summarized from images. Feature extractors were used to obtain this representative information.

Histogram: It is a way to quantify the intensity of pixels and cluster them groups representing ranges, called bins. As a result, a vector of accumulated intensities was obtained. It was used as it is a more quantifiable representation of a images, and further feature extractors can work over it, such as Skewness and Kurtosis.

Skewness: It is a measure of asymmetry of an image’s intensity distribution given the center as a reference point. It can be measured with the Fisher-Pearson coefficient as indicated in (2):

where is the mean intensity, is the standard deviation, and is the number of pixels.

Kurtosis: It is a measure of whether the data is heavy tailed or light-tailed distributed compared to a normal distribution. The formal definition is shown in (3):

where is the mean intensity, is the standard deviation, and is the number of pixels [11].

Haralick: Radiologists visually segregate different surfaces and its textures within the lungs to assess their condition, and whether there are uncommon conditions. Texture, an intrinsic property of a surface, contains important information about its structural composition. Therefore, feature extractors that can analyze such patterns, provide significant information for the stated problem. Haralick moments determine different patterns and their location throughout an image, by calculating statistical information associated to co-occurrence matrices, as shown in Fig. X. The elements of these matrices represent the probability of going from one pixel with intensityto another pixel of intensity , according to a certain distance and an angle of the neighborhood. By doing this, spatial distribution and levels of gray are obtained.

Chart, waterfall chart

Description automatically generated

Fig 1 Different levels of proximity for a pixel [12]

Hyperparameters used to tune this feature extractor were Blur and Distance. Blur smooths the image before the feature extraction. Distance determines the size of the neighborhood, by setting the maximum distance between two pixels that are considered neighbors.

Zernike: It captures information related to shape or geometry from an image. Zernike moments are resistant to rotation, non-redundant, and tolerant to noise. These moments are computed from the image’s intensity function projections on the orthogonal base functions. To do so, the image’s center is considered as the center of a unit disk, as shown in Fig. X. Each of the 64 moments are then calculated from the equations (4) and (5):

where will be the resultant moment, from the polynomial order and a given parameter .

A picture containing bubble chart

Description automatically generated

Fig 2 Radius size surrounding area of interest [13]

Hyperparameters used to tune this feature extractor were Degrees and Radius. Degrees determines the maximum degree of the Zernike moments that are calculated, as a result, the total number of them. Radius determines the size of the circle used to sample the image intensity values for computing the Zernike moments.

## E. Phase 1: ML-based Classification

Within a first approach in this research, feature extractors were used and their results feed into different classifiers to achieve the highest performance possible.

Support Vector Machine: SVM creates hyperplanes, which partition the data to classify it, as shown in Fig. X. SVMs were chosen as they allow to model highly complex relationships.



Fig 3 Data classification with hyperplane

It should be noted that SVM is a ML classifier that has shown a significant performance in similar task, which is the reason it was also tested in this study.

Random Forest: Also called Decision Tree Forests, combine bagging (generation of datasets by sampling original training data) and random feature selection to add diversity to the decision tree models. After trees are generated, the model uses a vote to combine trees’ predictions. It was chosen as an alternative model as they can handle large datasets, as only a small portion of the full feature set is used.

Boosting: It boosts performance of weak learners using models trained on resampled data and a weighted vote to determine the final prediction. Contrary to bagging, resampled datasets are constructed specifically to generate complementary learners. A common boosting algorithm is AdaBoost or adaptative boosting, which is tree-based implemented. This model can also effectively handle large datasets, and it was chosen for this reason.

## F. Phase 2: NN-based Classification

Posteriorly, Neuronal Networks (NNs) were an additional tool used, to achieve a better performance or as an alternative classification method.

NNs use artificial neurons, shown in Fig. X, as building blocks to model complex data. The neuron is modeled mathematically as indicated by (4):

where represents the value of a feature , and the weight given to each feature. A first strategy included training and using a simple NN as a classifier, with its inputs being the results obtained from feature extractors mentioned in Phase 1.



Fig 4 Typical artificial neuron

Convolutional Neuronal Networks (CNNs) are NNs that include at least one convolutional layer. A typical CNN consists of the combination of convolutional layers, pooling layers, and dense layers. A second strategy within this study was training a CNN with preprocessed images and posteriorly using it as a classifier. The architecture of this CNN was arbitrarily generated.

Pre-trained CNNs have been previously trained for other classification tasks, and have already learnt pattens and features, which can be reused for a different problem. When doing so, the last layers are usually changed to adapt them to a new output, while the weights obtained in the rest of the layers remain the same. For the scope of this study, a last strategy consisted of individually testing three pre-trained CNNs mentioned in the Literature: MobileNet V2 shown in Fig. X, Resnet18 and Resnet 50 shown in Fig. X.

A picture containing funnel chart

Description automatically generated

Table

Description automatically generated

Fig 5 MobileNet V2’s structure [14]

Table

Description automatically generated

Fig 6 Resnet50’s structure [15]

## G. Metrics

In order to measure the performance of different preprocessing techniques, data and classifiers, the metrics described below where used.

Confusion matrix: It is a table that categorizes predictions according to whether they belong or not to a class. The size of the matrix depends on the number of classes. The class of interest is known as positive class, and the others are negative. The matrix will be composed of the following categories, as shown in Fig. X for a two classes example:

* True Positive (TP): Correctly classified as class of interest
* True Negative (TN): Correctly classified as not class of interest
* False Positive (FP): Incorrectly classified as class of interest
* False Negative (FN): Incorrectly classified as not class of interest



Fig 7 Two-class confusion matrix

The importance of this matrix falls in all the performance metrics derived from it.

Accuracy: Also called success rate, It is defined by (5):

Precision: Also known as positive predictive value, measures the proportion of positive examples that are truly positive as defined by (6):

Sensitivity: Also called true positive rate, measures the proportion of positive examples correctly classified, as defined by (7):

Specificity: Also called true negative rate, measures the proportion of negative examples correctly classified, as defined by (8):

Balanced accuracy:

F measure: Also called F1-score, describes de model performance, by combining recall and precision, as defined in (10):

Matthews Correlation Coefficient: It is a statistical tool for model evaluation, is defined by (11):

Even though it is widely used, it does not perform well in imbalanced datasets [16]. This problem was tacked in the Data Splitting phase.

## H. Loss function and Optimizer

A loss function and an optimizer are required to handle the process of updating and choosing the best possible parameters for the classification model.

A loss function measures the difference between predicted and true probability distributions. Weighted cross entropy loss is a variation of cross entropy loss. It is especially useful for class imbalance, as it assigns different weights to different classes, as shown in (12) [17]:

where, is the weight assigned to the class , the true label (either 1 or 0), is the probability of the class , and is the total number of classes. For the scope of this study, weights were calculated using (13):

An optimizer is required to update the weights of a neuronal network. Adam optimizer avoids oscillations and convergence problems during training. It can be tuned with various parameters, as shown in Fig. X [18]. The following values for the hyperparameters of Adam were chosen Betas , and Learning rate .

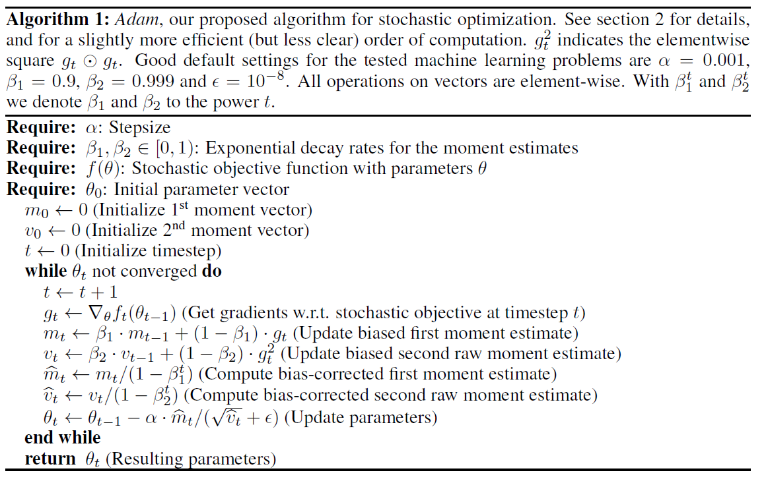


Fig 8 Adam algorithm

The methodology of this study is summarized in Table X.

Table 2 Methodology of Lung diseases classification study

|  |  |  |  |
| --- | --- | --- | --- |
| Test | **Phase** | **Feature extraction** | **Classifier** |
| 1 | 1 | Histogram, Skewness, Kurtosis, Haralick, Zernike, and possible combinations | SVM, RFT, Boosting |
| 2 | 2 | Neuronal Network |
| 3 | 2 | Convolutional Neuronal Network | |
| 4 | 2 | Pre-trained NN: MobileNet V2 | |
| 5 | 2 | Pre-trained NN: ResNet18 | |
| 6 | 2 | Pre-trained NN: ResNet50 | |

# III. Results and Discussion

*Phase 1 – Feature Extraction*

The relation between the model accuracy and the feature extractors is shown in Table X. Haralick and Zernike moments together outperform the rest of combinations tested, with a balanced accuracy of 88.03% for binary and 82.01% for multiclass classification. When both feature extractors work independently, the accuracy can reduce as much as 12% in the case of Zernike and binary classification.

Haralick and Zernike hyperparameters were optimized, as shown in Fig. 9. Variation of Blur, either including it or not, shows that including the total accuracy up to 0.6% for multiclass classification. Variation of Distance shows that a smaller distance will increase the total accuracy in up to 6.81% for binary and 1.51% for multiclass classification, when comparing values of 1 and 10. Variation of Degree shows that a smaller distance will increase the total accuracy in up to 10.58% for binary and 2.04% for multiclass classification, when comparing values of 8 and 20.

*Phase 1 – Classification*

SVM obtained a higher balanced accuracy when compared to other ML algorithms. The comparison was performed using Haralick and Zernike moments as feature extractors, and hyperparameters values Distance=1, Radius=180 and Degree=8. A difference of 16.89% for binary and 11.04% for multiclass classification, compared to Boosting, was obtained as shown in Table 4. Comparable results have been found in other researches.

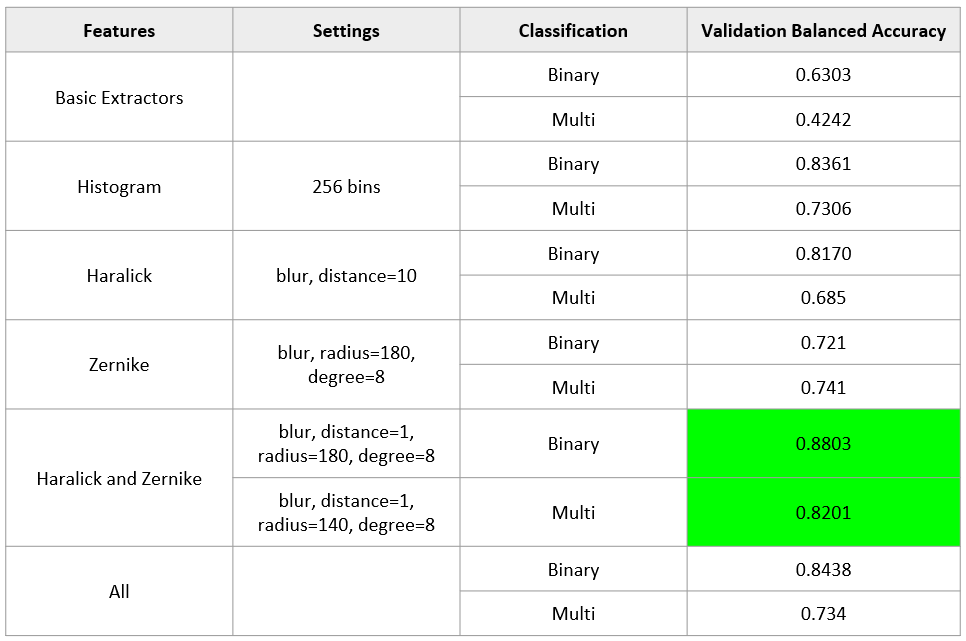
*Phase 2 - Classification*

Using an NN as a classifier, with its inputs being the feature extractors of Phase 1, Haralick and Zernike moments, produced slightly better results in terms of accuracy than SVM. A difference of 2.81% for binary and 0.52% for multiclass classification, compared to Boosting, was obtained as shown in Table 5. This can be explained due to the obtainable nonlinearity in NN.

Results using different DL models are shown in Table 5. A simple CNN, without Augmentation in the preprocessing phase, performed the best for binary classification in clean datasets with 92.75% of balanced accuracy, while MobileNet V2 did in multiclass classification with 88.92%. It can be observed that the randomness of Online Augmentation decreased the general performance of every model for clean datasets. ResNet50 provided the highest balanced accuracy of 85.21% for binary and 81.95% for multiclass classification. Offline Augmentation produces higher accuracy in every model when compared to Offline Augmentation. MobileNet V2 provided the highest balanced accuracy of 90.67% for binary and 87.59% for multiclass classification.

In the case of noisy images, Table 6 show the balanced accuracy obtained with different NNs. A simple CNN outperformed in binary classification with 84.39% of balanced accuracy. A simple NN, with Haralick and Zernike moments as feature extractors, outperformed in multiclass classification with 78.80% of balanced accuracy.

Table 3 Balanced accuracy related to feature extraction



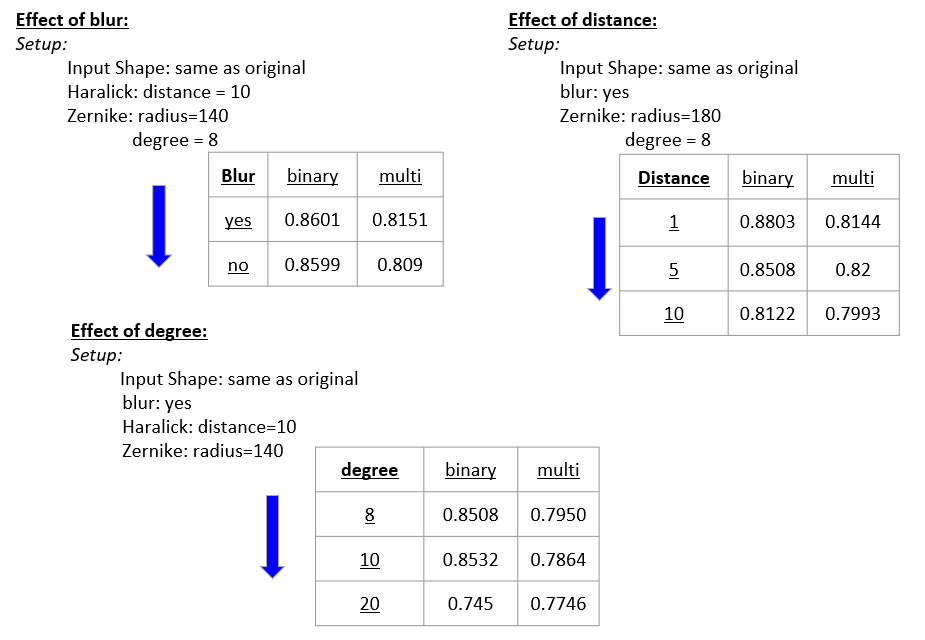


Fig 9 Hyperparameters variation on Haralick and Zernike

Table 4 Balanced accuracy comparison for ML Classifiers

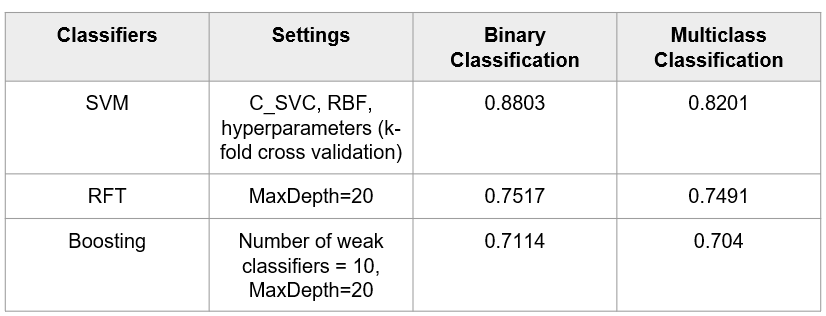


Table 5 Balanced accuracy comparison for DL Classifiers

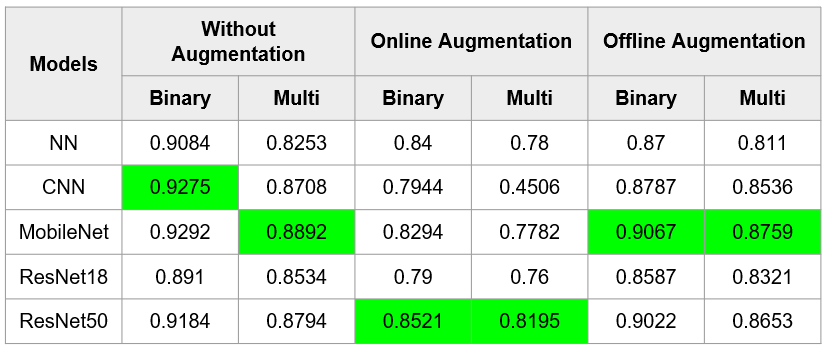
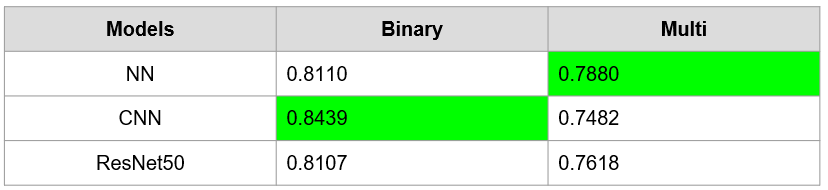


Table 6 Balanced accuracy using DL for noisy test dataset



# IV. Conclusions & Recommendations

Haralick and Zernike, together with SVM, performed the best for the objective of this study. Better performance can be achieved for Haralick and Zernike by using Blur as well as small values for the hyperparameters Distance and Degree.

The pretrained CNN MobileNet V2, combined with Online Augmentation, provided better results that the models tested with Offline Augmentation. A simple CNN together with a NN performed the best for a noisy dataset.

The present study only considered the last layer training of pre-trained CNNs. Therefore, training more layers can possibly improve the results, specifically when it comes to handling a noisy data.

Based on the literature review, quality of the CRX images of the training database must be observed to develop more accurate DL detection models. Also, additional steps such as lung segmentation or combination of pretrained NNs would allow more precise results.

Finally, the balanced accuracy obtained with the best models in this study are approximate to ones obtained in similar researches, with a difference no bigger than 10%.

References

[1] e. a. Nandhini Subramanian, “A review of deep learning-based detection methods for COVID-19,” 2022.

[2] Emre AVUÇLU, “COVID-19 detection using X-ray images and statistical measurements,” 2022.

[3] Juliana C. Gomes, Valter A. de F. Barbosa, et al., “IKONOS: an intelligent tool to support diagnosis of COVID-19 by texture analysis of X-ray images,” 2020.

[4] Geeta Rani, Ankit Misra, et al., “A multi-modal bone suppression, lung segmentation, and classification approach for accurate COVID-19 detection using chest radiographs,”

[5] S. L. Dongguang Li, “An artificial intelligence deep learning platform achieves high diagnostic accuracy for Covid-19 pneumonia by reading chest X-ray images,” 2022.

[6] ISM, *Chest X-Ray images dataset.* [Online]. Available: http://​gofile.me​/​5AIED/​xNqNoKPxu

[7] Angel Igareta, *Stratified Sampling: You May Have Been Splitting Your Dataset All Wrong*. Towards Data Science. [Online]. Available: https://​towardsdatascience.com​/​

[8] Brett Lantz, *Machine Learning with R,* 2nd ed. Packt Publishing, 2015.

[9] Sidheswar Routraya, Arun Kumar Raya, Chandrabhanu Mishrab, “Image denoising by preserving geometric components based on weighted bilateral filter and curvelet transform,” 2018.

[10] Mingle Xu, Sook Yoon, et al., “A Comprehensive Survey of Image Augmentation Techniques for Deep Learning,” 2023.

[11] NIST/SEMATECH, *Handbook of Statistical Methods*. NIST/SEMATECH. [Online]. Available: https://​www.itl.nist.gov​/​div898/​handbook/​index.htm

[12] *Haralick texture.* [Online]. Available: https://​cvexplained.wordpress.com​/​2020/​07/​22/​10-​6-​haralick-​texture/​

[13] *Zernike Moments.* [Online]. Available: https://​cvexplained.wordpress.com​/​2020/​07/​21/​10-​5-​zernike-​moments/​

[14] Pytorch Team, *Mobilenet V2.* [Online]. Available: https://​pytorch.org​/​hub/​pytorch\_​vision\_​mobilenet\_​v2/​

[15] Pytorch Team, *ResNet.* [Online]. Available: https://​pytorch.org​/​hub/​pytorch\_​vision\_​resnet/​

[16] Qiuming Zhu, “On the performance of Matthews correlation coefficient (MCC) for imbalanced dataset,” 2020.

[17] e. a. Sheng Lu, “Dynamic Weighted Cross Entropy for Semantic Segmentation with Extremely Imbalanced Data,” 2019.

[18] Diederik P. Kingma, Jimmy Lei Ba, “Adam: A method for Stochastic Optimization,” 2015.