COVID-19 Features Detection Using Machine Learning Models and Classifiers

Ali Al-Bayaty and Marek Perkowski

Abstract Different machine learning techniques and approaches were implemented to detect the features of COVID-19, from chest X-Ray and CT medical images, as well as to identify them from other similar human-being lungs infection diseases. In this work, Logistic Regression, Neural Networks, Random Forests, Decision Trees, kNN, and CN2 Rule Induction are the machine learning models and classifiers that were utilized to perform such detection and identification. The entire process according to the importance of good parameters selection, and such performance was presented and emphasized at different phases of models analysis and visualization. In our presented method, the achieved classification accuracies were up to 95.5%. Our work was implemented using Orange software, as a visual-based tool, and dedicated for physicians with no experience in machine learning algorithms and programming languages.

Keywords Coronavirus · COVID-19 · Features detection · Features extraction · Machine learning classifiers

1 Introduction

The Coronavirus disease pandemic, a.k.a. *COVID-19* by WHO (World Health Organization) [1], is world widely spread affecting many people in different countries. Since this virus does not have a well-known information as well as not match any similar symptoms that occur by other well-documented and knowledge-based viruses, medical concerns and emergency orders were firmly raised in many regions, e.g. schools and businesses closures as well as stay-home orders, to protect lives

A. Al-Bayaty (✉) · M. Perkowski
Electrical and Computer Eng. Dept., Portland State University, Portland, USA
e-mail: albayaty@pdx.edu

M. Perkowski
e-mail: h8mp@pdx.edu

S. Adibi et al. (eds.), *The Science behind the COVID Pandemic and Healthcare Technology Solutions*, Springer Series on Bio- and Neurosystems 15,
https://doi.org/10.1007/978-3-031-10031-4_18

and eliminate the disease outbreak excessively as well. For that, hourly data collections regarding infected areas, communities, and people are continuously gathered, accumulated, and compiled to form visual representations of COVID-19 morbidness cases and their spreading in different zones.

From the scientific and engineering point of view, the machine learning algorithms and models can play a role in classifying and identifying the COVID-19 cases from other similar human-being lungs infection diseases, such as Bacterial Pneumonia, Viral Pneumonia, Pneumocystis, Streptococcus, and SARS, using chest X-Ray and CT (Computerized Tomography) medical images. Some researchers were able to have fairly classified results from such diagnostic images using different models of ImageNet deep learning classifiers [2, 3], as described in [4–6]. Alimadadi et al. stated how governments, research institutes, and technological companies issued an urgent call-to-action for AI (Artificial Intelligence) researchers to develop data mining techniques and build open-source real-time analytical datasets to fight the COVID-19 pandemic and stop its spreading. Such that, the collection of large real-time diagnostic datasets of COVID-19 cases can give a better understanding of the COVID-19 patterns and spreading, as well as improve the speed and accuracy of the medical analyses when such diagnostic datasets are integrated with machine learning algorithms [7]. Pinter et al. mentioned that due to the lack of necessarily diagnostically datasets collection, the epidemiological models were in challenge in the matter of higher accuracy delivery for a long-term prediction. They implemented the ANFIS (Adaptive Neural-Fuzzy Inference System) and MLP-ICA (Multi-Layer Perceptron-Imperialist Competitive Algorithm) as a hybrid machine learning algorithms approach to predict the time-series of the infected individuals and their mortality rates, using MATLAB's ANFIS toolbox. Moreover, their work can be considered as an initial benchmarking tool for future research regarding the potential of machine learning algorithms in COVID-19 pandemic prediction [8]. In the paper of Elaziz et al., a new machine learning method was presented to classify the COVID-19 and non-COVID-19 cases using chest X-ray images. The features extracted from such images using the method of FrMEMs (Fractional Multichannel Exponent Moments) within a parallel workstation to accelerate the overall computational process. Their work was evaluated using two COVID-19 chest X-ray datasets that achieved accuracy rates of 96.09% and 98.09% for the first and second datasets, respectively [9]. Sujath et al. presented a model to predict the spread of COVID-19 in India, this approach was implemented using different machine learning models, such as Linear Regression, MLP, and Vector Autoregression using the COVID-19 Kaggle repository datasets for the epidemiological COVID-19 cases in India, only. Their work showed that the CI (Confidence Interval), a way of quantifying the uncertainty of estimated results, was 95% for the completely implemented machine learning models [10]. Ardabili et al. presented a comparative analysis of different machine learning models, such as MLP and ANFIS, and soft computing models, such as GA (Genetic Algorithm), PSO (Particle Swarm Optimization), and GWO (Grey Wolf Optimizer), to predict the COVID-19 outbreak and spreading. They demonstrated that the machine learning algorithms were effective tools and have more promising results than the soft computing models in the COVID-19 pandemic prediction [11]. Brinati et al. described the amplification of

COVID-19 viral RNA using the rRT–PCR (real-time Reverse Transcription–Polymerase Chain Reaction) is the current gold standard test for COVID-19 infections confirmation. However, due to the rRT-PCR weaknesses, such as false-negative rates of 15–20%, and a potential shortage of reagents, therefore faster, less expensive, and more accessible testing methods, as alternative solutions, should be developed. Two machine learning models were implemented to classify patients as either positive or negative to the COVID-19 infection using the hematochemical values from the routine blood exams. In such two models, the accuracy ranges 82–86% and the sensitivity ranges 92–95% with respect to the gold standard test. Moreover, a Decision Tree model was implemented as straightforward simple decision assistance for the COVID-19 suspected cases [12]. Cheng et al. indicated that approximately 20–30% of COVID-19 infected cases need hospitalization, while 5–12% of them may require critical care in the ICU (Intensive Care Unit). In their work, they developed a machine learning algorithm, as a risk prioritization tool, using Random Forest model to predict the ICU requirements within the 24-h. Time-series information, laboratory data, vital signs, nursing valuations, and ECG (Electrocardiograms) signals were used as input datasets for this model. These datasets were randomly split into 70% of the training dataset and 30% of the test dataset. Then, this model was trained using the tenfold CV (Cross-Validation) technique. The model's performance and prediction was evaluated on the test dataset, as the following: sensitivity of 72.8% (95% CI: 63.2–81.1%), specificity of 76.3% (95% CI: 74.7–77.9%), accuracy of 76.2% (95% CI: 74.6–77.7%), and Area under ROC of 79.9% (95% CI: 75.2–84.6%). Thus, this machine learning tool could improve the planning and the management of hospital resources in more effective ways regarding the COVID-19 patients' hospitalization [13]. In the study of Rustam et al., different supervised machine learning models were implemented to forecast the number of patients infected by COVID-19. Four standard models, such as Linear Regression, Exponential Smoothing, LASSO (Least Absolute Shrinkage and Selection Operator), and SVM (Support Vector Machine), were developed to forecast the upcoming patients and results. The number of newly infected patients, the number of recovered patients in the next 10 days, and the number of deaths were the three prediction categories for each model. The Exponential Smoothing model had better forecasting results than the other models, followed by the Linear Regression model, and then the LASSO model performed well in forecasting the newly infected cases, recovery rate, and death rate. While the SVM model performed poorly in the three prediction categories. The three models were performed on the Johns Hopkins University's COVID-19 repository datasets [14].

2 Materials and Methods

A large set of diagnostic medical images of human-being lungs diseases, i.e. *datasets*, has to be publicly provided to the scientific communities, to achieve the valuable comparable results of identification and classification of COVID-19 cases. For this reason, many medical datasets [15, 16] have been already published to stop the

COVID-19 outbreak using vast sets of technologies and applied approaches, e.g. machine learning algorithms. The aim of our work is to identify the best classifiers, within the best CA (Classification Accuracy) scores, that classify the COVID-19 infected cases from other human-being lungs infected cases. As well as to provide a convenient visual-based classification tool to physicians, without their need to understand the deep knowledge of machine learning algorithms, the programming languages, such as Python, the computational libraries, such as NumPy, Pandas, and scikit-learn, and the abstraction platforms, such as PyTorch and TensorFlow. Our work was fully designed, built, and implemented using the open-source machine learning and data visualization tool *Orange*, from the University of Ljubljana [17].

Orange provides data analysis, data visualization, statistical distributions, and vast sets of plotting tools, as well as its GUI (Graphical User Interface) allows the physicians to focus on data analysis and manipulating, instead of coding, to accelerate the diagnostically identification and classification workflows with ease. Our presented work with Orange was performed using the datasets that were provided by the Kaggle repository [15], and the workflow was categorized into four proposed phases: (1) Datasets Preparation Phase, (2) Training Dataset Operations Phase, (3) Test Dataset Operations Phase, and (4) Prediction and Performance Phase. Furthermore, seven machine learning algorithms were utilized, in this work, to have sufficient comparable decisions regarding the best-chosen model for the best classification accuracy to identify and classify the COVID-19 cases. These models are:

- *Distances* that computes the distances between the rows and columns in the datasets, i.e. the cases [18, 19],
- *Logistic Regression* that classifies the cases using the non-linear Sigmoid function (σ) [19, 20],
- *CN2 Rule Induction* that uses efficient induction of simple and comprehensive rules in the form of (IF *condition* THEN *predict case*) to predict the cases [21, 22],
- *Tree* that splits the cases into nodes and leaves by labeling purity and forward pruning [23, 24],
- *Random Forest* that classifies the cases using an ensemble of decision trees [25, 26],
- *kNN* (*k*-Nearest Neighbors) that searches for the *k* closest cases based on their features and their averages as classification factors [27, 28], and
- *Neural Network* that classifies the cases within the MLP model using the backpropagation method [29].

Note that all the aforementioned algorithms are supervised classifiers, except the *Distances*, which is an unsupervised classifier.

Note that some of Orange's toolboxes have been renamed in purpose to match their underlying phases as well as their designated operations, for ease of follow and understanding.

2.1 Datasets Preparation Phase

The datasets were gathered from the chest X-Ray (of PA “Posteroanterior”, AP “Anteroposterior”, APS “AP Supine”, and L “Lateral” captures [30]) and CT (of Axial and Coronal scans [31]) medical images of COVID-19, SARS, Pneumocystis, and Streptococcus infection cases. Since the number of these gathered images is relatively small, the data augmentation technique is performed in this phase. The data augmentation was achieved by enlarging the size of each medical image, i.e. *sample*, in the datasets by increasing their (1) *brightness* by the factor of 16 and (2) *contrast* by the factor of 32, to have a sufficient number of samples. A sufficient number of samples yields a good identification and better classification accuracy judgment, as well as fulfills Hoeffding’s Inequality generalization bound [32]. For instance, to demonstrate Hoeffding’s Inequality, for any randomly selected size of N COVID-19 and non-COVID-19 samples, the generalization bound for the probability ($P[\cdot]$) of such an event for any tolerance ($\epsilon > 0$) is as stated in Eq. (1):

$$P[|v - \mu| > \epsilon] \leq 2e^{-2\epsilon^2 N} \tag{1}$$

where, μ is the probability of the COVID-19 samples in a bin consisting of COVID-19 and non-COVID-19 samples, while v is the fraction of the selected COVID-19 samples among the non-COVID-19 samples. Note that Hoeffding’s Inequality formula mostly affects by how large N and the chosen ϵ are.

The ACDSee® photo editing software was used to implement such data augmentation [33]. Note that other data augmentation methods, such as shape sheering, angles rotation, and flipping/mirroring, were not used in this phase, due to the fact that physicians usually check such diagnostic images in the normal straightway position, as portraits. For fast data processing, the samples were scaled uniformly to the dimensions of 128 × 128 pixels, and the resultant statistics of these samples are summarized in Table 1.

Figure 1 demonstrates the preparation phase of these datasets that consists of the following Orange toolboxes: (1) *Import Images* that loads the datasets locally, i.e. from the computer, (2) *Image Viewer* that visually checks the loaded datasets, (3) *Adding Labels* that generates the cases for each sample. Since these datasets are non-labeled and most of the classifiers are supervised, then a column of labels, i.e. *classes*, was generated regarding the samples’ filenames. Note that, each filename

Table 1 Infected samples statistics

Infection case	Number of samples
COVID-19	169
SARS	33
Pneumocystis	45
Streptococcus	51
Total =	298

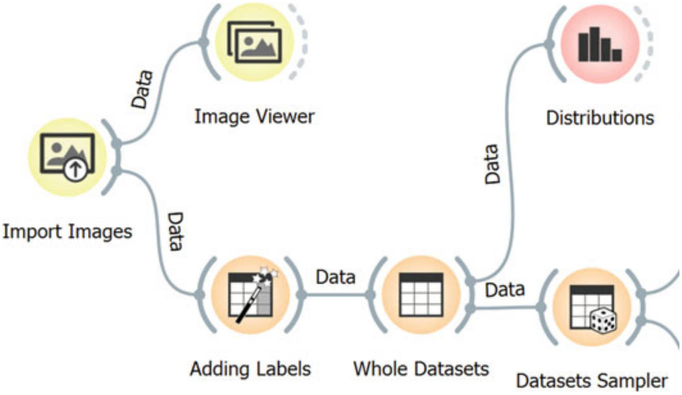
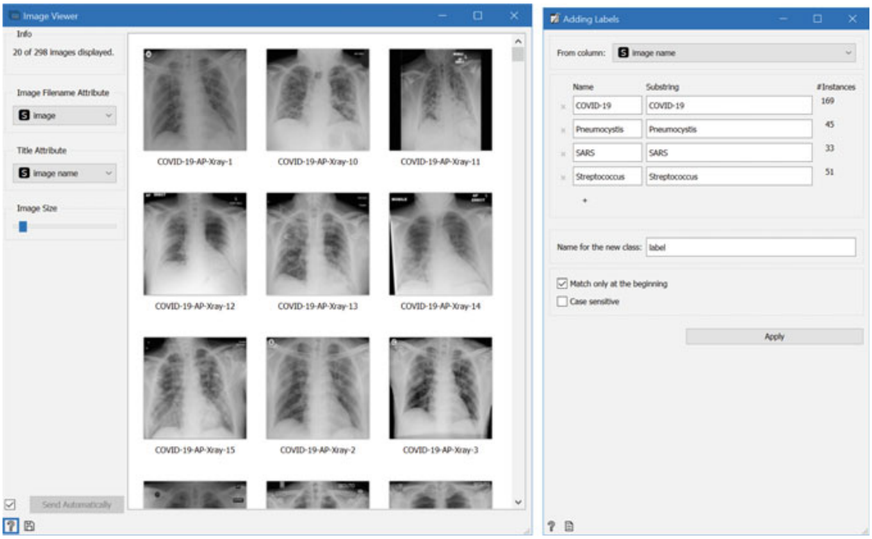


Fig. 1 Datasets preparation phase using orange toolboxes

corresponds to an infected case name, for instance the filename of the "COVID-19-APSupine-Xray-1.jpeg" sample generates the "COVID-19" case as a label. (4) *Whole Datasets* that lists the whole labeled datasets, (5) *Distributions* that statistically displays the labeled datasets, and (6) *Datasets Sampler* that samples the whole labeled datasets into 70% of training dataset and 30% of test dataset, i.e. 209 training samples and 89 test samples.

For more illustration, Fig. 2 illustrates the *Image Viewer* and *Adding Labels* toolboxes of Orange.



Info
20 of 298 images displayed.

Image Filename Attribute: image

Title Attribute: image name

Image Size

☑ Send Automatically

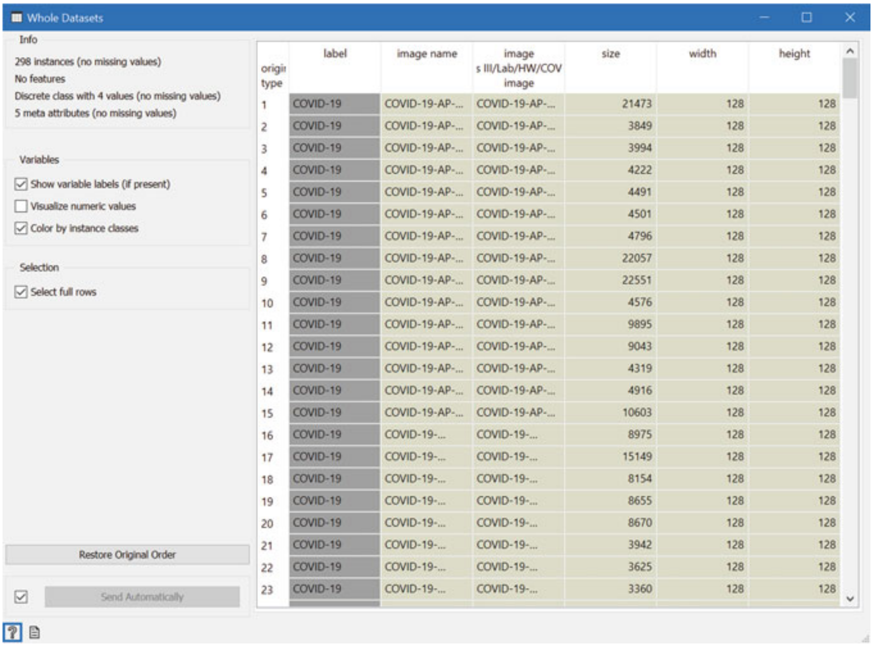
From column: image name

Name	Substring	#Instances
COVID-19	COVID-19	169
Pneumocystis	Pneumocystis	45
SARS	SARS	33
Streptococcus	Streptococcus	51

Name for the new class: label

☑ Match only at the beginning
☐ Case sensitive

Fig. 2 Orange image viewer (left) and adding labels (right) toolboxes



Info

298 instances (no missing values)
No features
Discrete class with 4 values (no missing values)
5 meta attributes (no missing values)

Variables

☑ Show variable labels (if present)
□ Visualize numeric values
☑ Color by instance classes

Selection

☑ Select full rows

origi type	label	image name	image s III/Lab/HW/COV image	size	width	height
1	COVID-19	COVID-19-AP-...	COVID-19-AP-...	21473	128	128
2	COVID-19	COVID-19-AP-...	COVID-19-AP-...	3849	128	128
3	COVID-19	COVID-19-AP-...	COVID-19-AP-...	3994	128	128
4	COVID-19	COVID-19-AP-...	COVID-19-AP-...	4222	128	128
5	COVID-19	COVID-19-AP-...	COVID-19-AP-...	4491	128	128
6	COVID-19	COVID-19-AP-...	COVID-19-AP-...	4501	128	128
7	COVID-19	COVID-19-AP-...	COVID-19-AP-...	4796	128	128
8	COVID-19	COVID-19-AP-...	COVID-19-AP-...	22057	128	128
9	COVID-19	COVID-19-AP-...	COVID-19-AP-...	22551	128	128
10	COVID-19	COVID-19-AP-...	COVID-19-AP-...	4576	128	128
11	COVID-19	COVID-19-AP-...	COVID-19-AP-...	9895	128	128
12	COVID-19	COVID-19-AP-...	COVID-19-AP-...	9043	128	128
13	COVID-19	COVID-19-AP-...	COVID-19-AP-...	4319	128	128
14	COVID-19	COVID-19-AP-...	COVID-19-AP-...	4916	128	128
15	COVID-19	COVID-19-AP-...	COVID-19-AP-...	10603	128	128
16	COVID-19	COVID-19-...	COVID-19-...	8975	128	128
17	COVID-19	COVID-19-...	COVID-19-...	15149	128	128
18	COVID-19	COVID-19-...	COVID-19-...	8154	128	128
19	COVID-19	COVID-19-...	COVID-19-...	8655	128	128
20	COVID-19	COVID-19-...	COVID-19-...	8670	128	128
21	COVID-19	COVID-19-...	COVID-19-...	3942	128	128
22	COVID-19	COVID-19-...	COVID-19-...	3625	128	128
23	COVID-19	COVID-19-...	COVID-19-...	3360	128	128

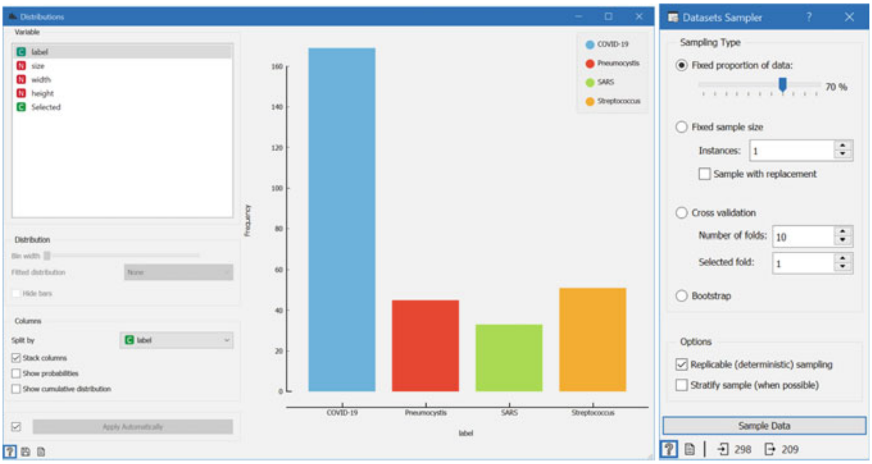
Fig. 3 Orange whole datasets toolbox

Note that four labels were generated using *Adding Labels* toolbox with labels as *COVID-19*, *Pneumocystis*, *SARS*, and *Streptococcus*. Figure 3 shows the *Whole Datasets* toolbox of Orange

Finally, Fig. 4 demonstrates the *Distributions* and *Datasets Sampler* toolboxes of Orange.

2.2 *Training Dataset Phase*

After the 209 samples, i.e. the training dataset, were received from the *Datasets Sampler* toolbox from the Datasets Preparation Phase, they can be buffered and visually checked along with their generated labels using the *Training Dataset* toolbox. Since these training samples contain no useful classifiable information, the *Inception v3 Model (Training)* toolbox is in need to represent this training dataset into its vectorized equivalent representations, a.k.a. *features*. This toolbox generates 2048 numerical features for each sample that makes the next classification processes more meaningful, and such features can be viewed using the *Training Dataset Features* toolbox of Orange. Note that the *Inception v3 Model (Training)* toolbox is based on Google's Inception v3 CNN (Convolutional Neural Network) architecture [34] that is trained on ImageNet.



☑ Stack columns
□ Show probabilities
□ Show cumulative distribution

Sampling Type
● Fixed proportion of data: 70 %
○ Fixed sample size
Instances: 1
□ Sample with replacement
○ Cross validation
Number of folds: 10
Selected fold: 1
○ Bootstrap

Options
☑ Replicable (deterministic) sampling
□ Stratify sample (when possible)

Fig. 4 Orange distributions (left) and datasets sampler (right) toolboxes, the sampling percentage can be changed based on the model design and technical requirements

Then, these features along with their labels from the *Inception v3 Model (Training)* toolbox were fed to various machine learning classifiers for further identification and classification of training dataset based on their features (and labels, if they were supervised classifiers), as shown in Fig. 5. Note that, the calculated distances in the *Distances* classifier were visualized using the *Hierarchical Clustering* toolbox along with the *Hierarchical Clustering Viewer* toolbox, the generated rules in the *CN2 Rule Induction* classifier were viewed through the *CN2 Rule Viewer* toolbox, the generated tree with nodes and leaves in the *Tree* classifier were visualized using the *Tree Viewer* toolbox.

2.3 Test Dataset Phase

After the 89 samples, i.e. the test dataset, were received from the *Datasets Sampler* toolbox from the Datasets Preparation Phase, they can be buffered and visually checked along with their labels using the *Test Dataset* toolbox, as shown in Fig. 6. Since these test samples contain no useful classifiable information, the *Inception v3 Model (Test)* toolbox was applied here, and their features can be viewed within the *Test Dataset Features* toolbox.

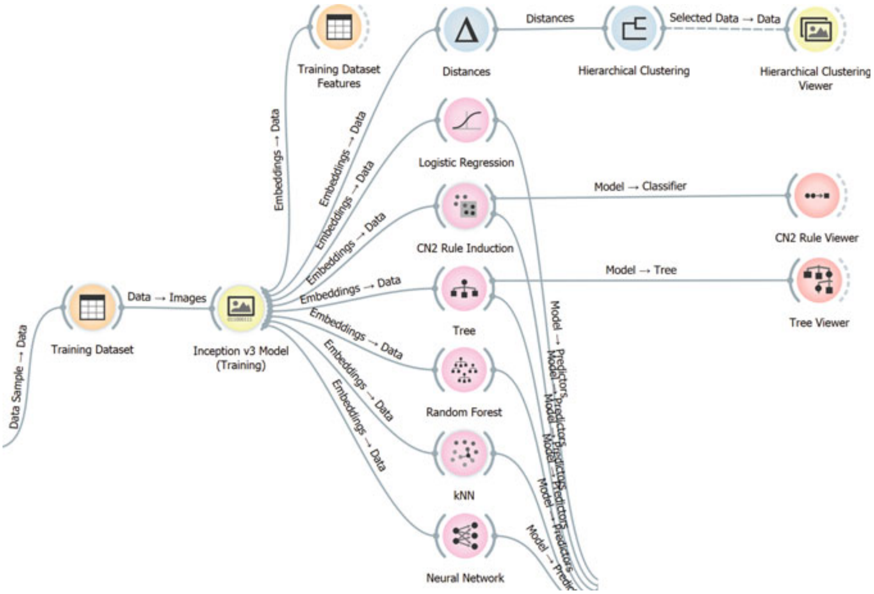


Fig. 5 Training Dataset Operations Phase using Orange toolboxes, (left) the training dataset was fed from the first phase then propagated to the Inception v3 Model for features generating, (middle) predictions were calculated from the machine learning classifiers then forwarded to the fourth phase for classification accuracies, (right) and the distances, rules, and trees can be viewed through the viewers

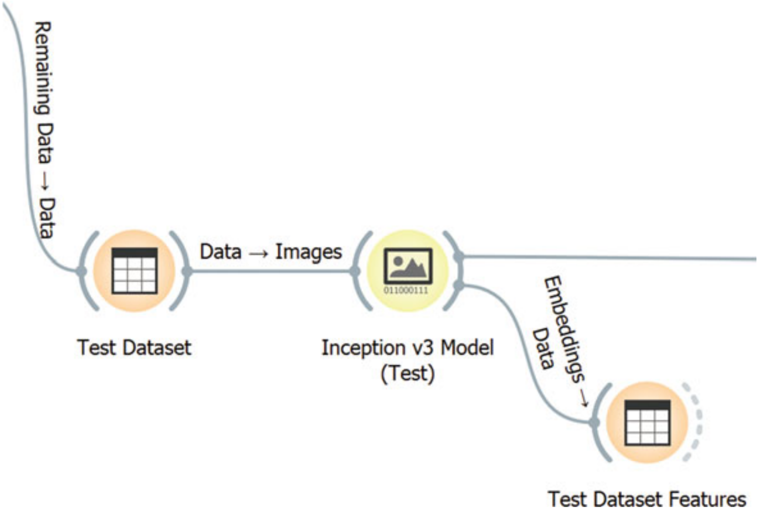


Fig. 6 Test Dataset Operations Phase using Orange toolboxes, (left) the test dataset was fed from the first phase, (middle) then propagated to the Inception v3 Model for features generating that fed to the fourth phase for performance measurement, (right) and these features can be visually viewed using test Dataset Features toolbox

2.4 Prediction and Performance Phase

The received signals from the Training Dataset Phase, i.e. the predictions from the machine learning classifiers, as well as from the Test Dataset Phase, i.e. the features from the *Inception v3 Model (Test)* toolbox were fed to the *Predictions* toolbox, to measure the performance and score the classification accuracies regarding the best-chosen classifier for COVID-19 cases. The following Orange toolboxes were implemented to compute such measurement and scoring, as illustrated in Fig. 7:

- *Confusion Matrix*: Shows the numbers of matched and unmatched samples from the test dataset under the predicted and the actual cases that were judged by the *Predictions* toolbox [35].
- *Sieve Diagram*: Visualizes the frequencies of cases based on a pair of classifiers [36].
- *Linear Projection*: Plots the linear separation of cases concerning their classifiers [37].

3 Methodology

In our presented work, an appropriate visual-based classification tool is provided that targets the medical domain, especially for the physicians with less experience in the machine learning philosophy and limited programming skills. Different distances, rules, trees, tables, and plots were obtained from the proposed seven machine learning classifiers based on their different selection of parameters. For that, the following

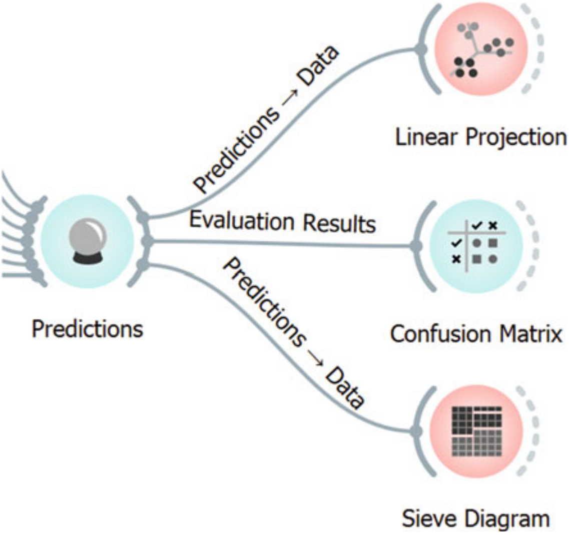
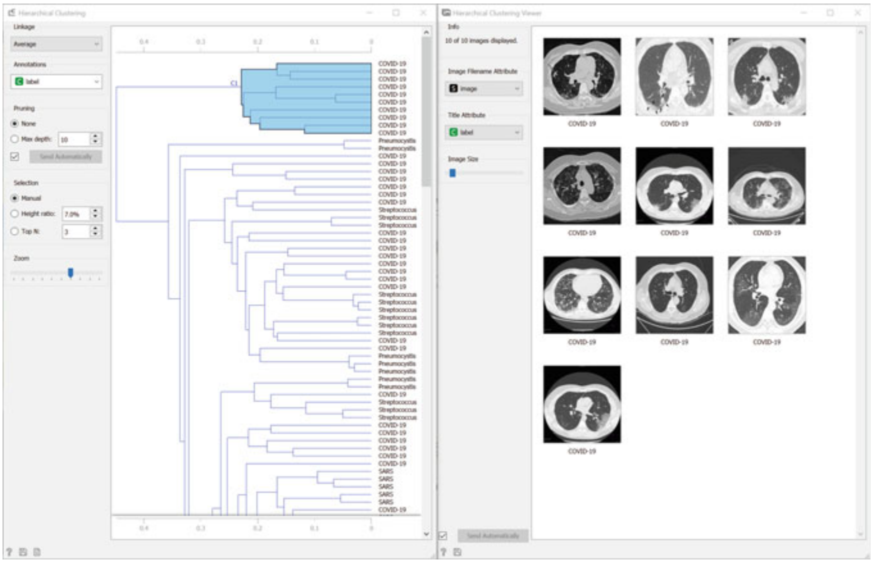


Fig. 7 Prediction and Performance Phase using Orange toolboxes, (left) the signals were received from the second and third phases, (right) and then the performance measurement and classification accuracies scoring were achieved through these three toolboxes


Hierarchical Clustering

Linkage
Average

Annotations
label

Pruning
● None
○ Max depth: 10
☑ Send Automatically

Selection
● Manual
○ Height ratio: 7.0%
○ Top N: 3

Zoom

Hierarchical Clustering Viewer

Info
10 of 10 images displayed.

Image Filename Attribute
image

Title Attribute
label

Image Size

☑ Send Automatically

Fig. 8 Distances and hierarchical clustering results of COVID-19 axial CT scans of 100% accuracy

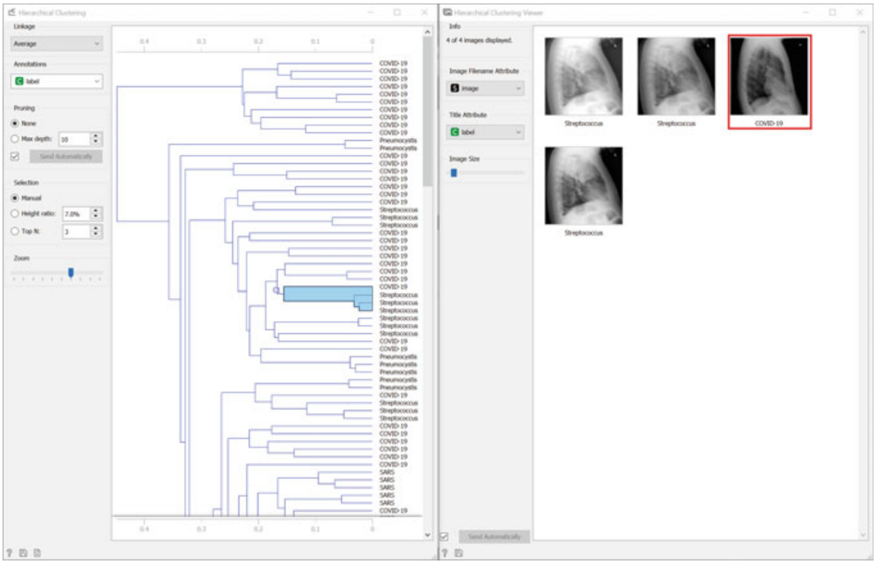
configurable parameters for each machine learning classifier were chosen and tuned to have the best human-classifiable results regarding COVID-19 infection cases among other similar human-being lungs infection cases. Note that some parameters have an insignificant effect or no effect at all on the classification results for some classifiers.

3.1 Parameters of Distances Classifier

The *Distances* toolbox, along with the *Hierarchical Clustering* and *Hierarchical Clustering Viewer* toolboxes, performs a good classification on the training dataset with a small number of errors in clustering the cases, as demonstrated in Figs. 8, 9, and 10. Few cases in the clustering process have mismatched results, and the explanations of these outcomes were left to the epidemiology specialists due to their deep knowledge in this field. Our work provides an adequate and easy-to-use tool for them. Note that the *Distance Metric* parameter is better to be set as *Cosine* when dealing with images, in general.

3.2 Parameters of CN2 Rule Induction Classifier

The *CN2 Rule Induction* toolbox produces different rules, in the format of (IF *condition* THEN *predict case*), and classification accuracies based on its *Evaluation measure* parameter. Such that, when this parameter was set to *Entropy* [38] its


Hierarchical Clustering

Linkage: Average

Annotations: label

Pruning
● None
○ Max depth: 10

☑ Send Automatically

Selection
● Manual
○ Height ratio: 7.0%
○ Top N: 3

Zoom

Hierarchical Clustering Viewer

Info
4 of 4 images displayed.

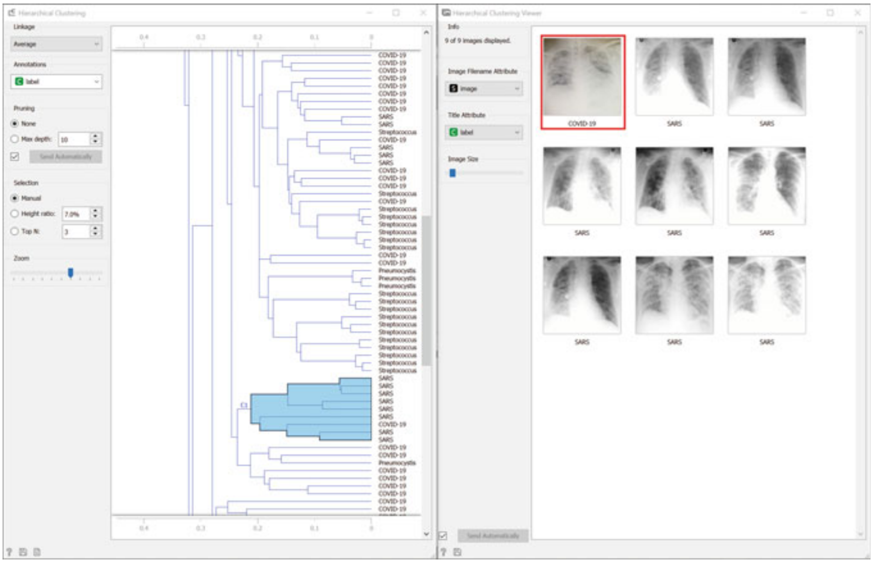
Image Filename Attribute: image

Title Attribute: label

Image Size

☑ Send Automatically

Fig. 9 Distances and hierarchical clustering results of streptococcus L X-Ray images with one mismatched case


Hierarchical Clustering

Linkage: Average

Annotations: label

Pruning
● None
○ Max depth: 10

☑ Send Automatically

Selection
● Manual
○ Height ratio: 7.0%
○ Top N: 3

Zoom

Hierarchical Clustering Viewer

Info
9 of 9 images displayed.

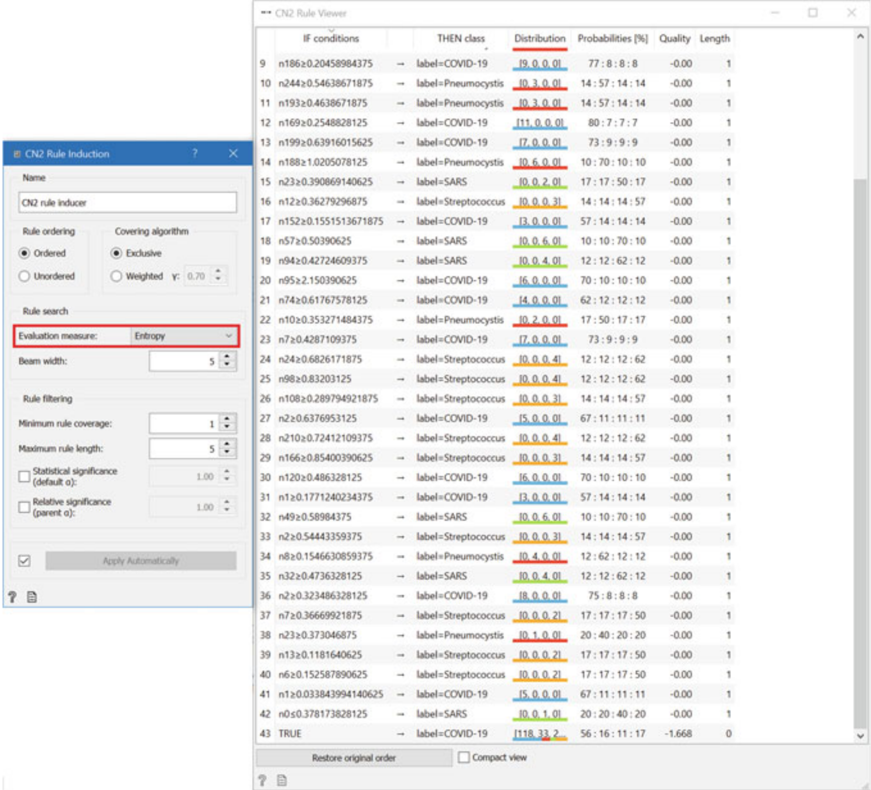
Image Filename Attribute: image

Title Attribute: label

Image Size

☑ Send Automatically

Fig. 10 Distances and hierarchical clustering results of SARS PA X-Ray images with one mismatched case



CN2 Rule Induction

Name: CN2 rule inducer

Rule ordering: ● Ordered ○ Unordered

Covering algorithm: ● Exclusive ○ Weighted γ: 0.70

Rule search

Evaluation measure: Entropy

Beam width: 5

Rule filtering

Minimum rule coverage: 1

Maximum rule length: 5

□ Statistical significance (default α): 1.00

□ Relative significance (parent α): 1.00

☑ Apply Automatically

CN2 Rule Viewer

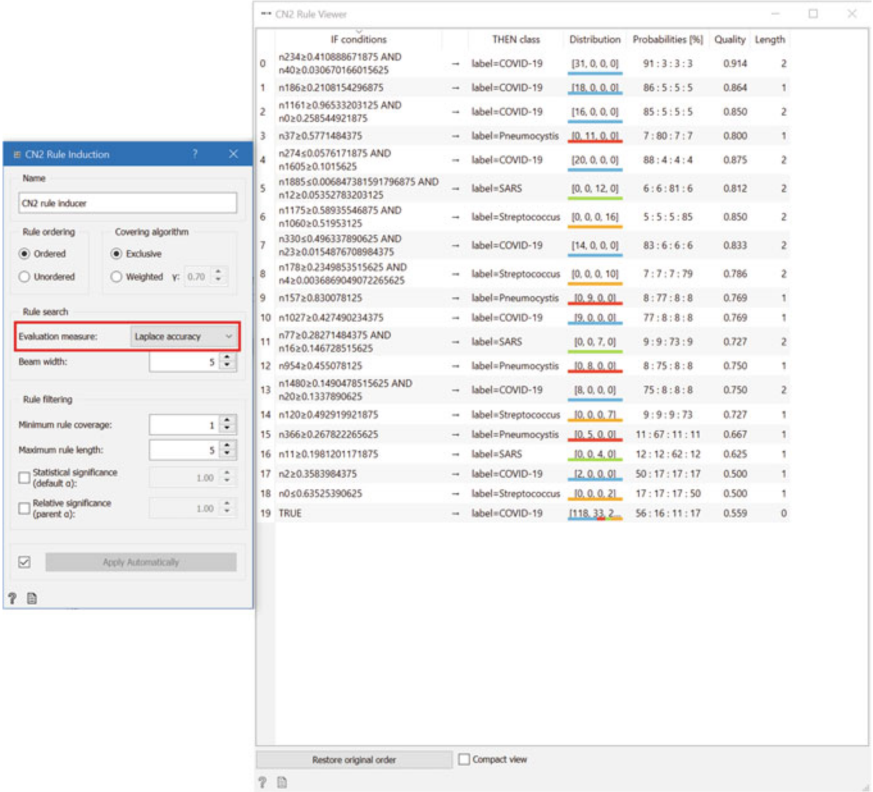
	IF conditions		THEN class	Distribution	Probabilities [%]	Quality	Length
9	n186≥0.20458984375	→	label=COVID-19	[9, 0, 0, 0]	77 : 8 : 8 : 8	-0.00	1
10	n244≥0.54638671875	→	label=Pneumocystis	[0, 3, 0, 0]	14 : 57 : 14 : 14	-0.00	1
11	n193≥0.4638671875	→	label=Pneumocystis	[0, 3, 0, 0]	14 : 57 : 14 : 14	-0.00	1
12	n169≥0.2548828125	→	label=COVID-19	[11, 0, 0, 0]	80 : 7 : 7 : 7	-0.00	1
13	n199≥0.63916015625	→	label=COVID-19	[7, 0, 0, 0]	73 : 9 : 9 : 9	-0.00	1
14	n188≥1.0205078125	→	label=Pneumocystis	[0, 6, 0, 0]	10 : 70 : 10 : 10	-0.00	1
15	n23≥0.390869140625	→	label=SARS	[0, 0, 2, 0]	17 : 17 : 50 : 17	-0.00	1
16	n12≥0.36279296875	→	label=Streptococcus	[0, 0, 0, 3]	14 : 14 : 14 : 57	-0.00	1
17	n152≥0.1551513671875	→	label=COVID-19	[3, 0, 0, 0]	57 : 14 : 14 : 14	-0.00	1
18	n57≥0.50390625	→	label=SARS	[0, 0, 6, 0]	10 : 10 : 70 : 10	-0.00	1
19	n94≥0.42724609375	→	label=SARS	[0, 0, 4, 0]	12 : 12 : 62 : 12	-0.00	1
20	n95≥2.150390625	→	label=COVID-19	[6, 0, 0, 0]	70 : 10 : 10 : 10	-0.00	1
21	n74≥0.61767578125	→	label=COVID-19	[4, 0, 0, 0]	62 : 12 : 12 : 12	-0.00	1
22	n10≥0.353271484375	→	label=Pneumocystis	[0, 2, 0, 0]	17 : 50 : 17 : 17	-0.00	1
23	n7≥0.4287109375	→	label=COVID-19	[7, 0, 0, 0]	73 : 9 : 9 : 9	-0.00	1
24	n24≥0.6826171875	→	label=Streptococcus	[0, 0, 0, 4]	12 : 12 : 12 : 62	-0.00	1
25	n98≥0.83203125	→	label=Streptococcus	[0, 0, 0, 4]	12 : 12 : 12 : 62	-0.00	1
26	n108≥0.289794921875	→	label=Streptococcus	[0, 0, 0, 3]	14 : 14 : 14 : 57	-0.00	1
27	n2≥0.6376953125	→	label=COVID-19	[5, 0, 0, 0]	67 : 11 : 11 : 11	-0.00	1
28	n210≥0.72412109375	→	label=Streptococcus	[0, 0, 0, 4]	12 : 12 : 12 : 62	-0.00	1
29	n166≥0.85400390625	→	label=Streptococcus	[0, 0, 0, 3]	14 : 14 : 14 : 57	-0.00	1
30	n120≥0.486328125	→	label=COVID-19	[6, 0, 0, 0]	70 : 10 : 10 : 10	-0.00	1
31	n1≥0.1771240234375	→	label=COVID-19	[3, 0, 0, 0]	57 : 14 : 14 : 14	-0.00	1
32	n49≥0.58984375	→	label=SARS	[0, 0, 6, 0]	10 : 10 : 70 : 10	-0.00	1
33	n2≥0.54443359375	→	label=Streptococcus	[0, 0, 0, 3]	14 : 14 : 14 : 57	-0.00	1
34	n8≥0.1546630859375	→	label=Pneumocystis	[0, 4, 0, 0]	12 : 62 : 12 : 12	-0.00	1
35	n32≥0.4736328125	→	label=SARS	[0, 0, 4, 0]	12 : 12 : 62 : 12	-0.00	1
36	n2≥0.323486328125	→	label=COVID-19	[8, 0, 0, 0]	75 : 8 : 8 : 8	-0.00	1
37	n7≥0.36669921875	→	label=Streptococcus	[0, 0, 0, 2]	17 : 17 : 17 : 50	-0.00	1
38	n23≥0.373046875	→	label=Pneumocystis	[0, 1, 0, 0]	20 : 40 : 20 : 20	-0.00	1
39	n13≥0.1181640625	→	label=Streptococcus	[0, 0, 0, 2]	17 : 17 : 17 : 50	-0.00	1
40	n6≥0.152587890625	→	label=Streptococcus	[0, 0, 0, 2]	17 : 17 : 17 : 50	-0.00	1
41	n1≥0.033843994140625	→	label=COVID-19	[5, 0, 0, 0]	67 : 11 : 11 : 11	-0.00	1
42	n0≤0.378173828125	→	label=SARS	[0, 0, 1, 0]	20 : 20 : 40 : 20	-0.00	1
43	TRUE	→	label=COVID-19	[118, 33, 2...	56 : 16 : 11 : 17	-1.668	0

Fig. 11 CN2 Rule Induction and Evaluation measure parameter as Entropy, lower classification accuracy and higher induced difficult-to-read rules

classification accuracy was 53.9% and the generated number of rules was 43, as shown in Fig. 11. However, when this parameter was set to *Laplace accuracy* [38], its classification accuracy changed to 64% and the generated number of rules to 19, as shown in Fig. 12. Therefore, the *Laplace accuracy* parameter produces more classification accuracy results with fewer easy-to-read induced classifiable rules. Note that, the colored bars in the *Distribution* column represent the infection case: Blue for COVID-19, Green for SARS, Orange for Streptococcus, and Red for Pneumocystis.

3.3 Parameters of Random Forest Classifier

The parameter *Number of trees* does not influence at all on the *Random Forest* toolbox CA performance and its results. Such that, when this parameter was set to 10, its CA is 70.8%. While, when this parameter was set to 20, its CA still be at 70.8%.



CN2 Rule Induction

Name: CN2 rule inducer

Rule ordering: ● Ordered ○ Unordered

Covering algorithm: ● Exclusive ○ Weighted γ: 0.70

Rule search

Evaluation measure: Laplace accuracy

Beam width: 5

Rule filtering

Minimum rule coverage: 1

Maximum rule length: 5

□ Statistical significance (default α): 1.00

□ Relative significance (parent α): 1.00

☑ Apply Automatically

CN2 Rule Viewer

	IF conditions		THEN class	Distribution	Probabilities [%]	Quality	Length
0	n234≥0.410888671875 AND n40≥0.030670166015625	→	label=COVID-19	[31, 0, 0, 0]	91 : 3 : 3 : 3	0.914	2
1	n186≥0.2108154296875	→	label=COVID-19	[18, 0, 0, 0]	86 : 5 : 5 : 5	0.864	1
2	n1161≥0.96533203125 AND n0≥0.258544921875	→	label=COVID-19	[16, 0, 0, 0]	85 : 5 : 5 : 5	0.850	2
3	n37≥0.5771484375	→	label=Pneumocystis	[0, 11, 0, 0]	7 : 80 : 7 : 7	0.800	1
4	n274≤0.0576171875 AND n1605≥0.1015625	→	label=COVID-19	[20, 0, 0, 0]	88 : 4 : 4 : 4	0.875	2
5	n1885≤0.006847381591796875 AND n12≥0.05352783203125	→	label=SARS	[0, 0, 12, 0]	6 : 6 : 81 : 6	0.812	2
6	n1175≥0.58935546875 AND n1060≥0.51953125	→	label=Streptococcus	[0, 0, 0, 16]	5 : 5 : 5 : 85	0.850	2
7	n330≤0.496337890625 AND n23≥0.0154876708984375	→	label=COVID-19	[14, 0, 0, 0]	83 : 6 : 6 : 6	0.833	2
8	n178≥0.2349853515625 AND n4≥0.0036869049072265625	→	label=Streptococcus	[0, 0, 0, 10]	7 : 7 : 7 : 79	0.786	2
9	n157≥0.830078125	→	label=Pneumocystis	[0, 9, 0, 0]	8 : 77 : 8 : 8	0.769	1
10	n1027≥0.427490234375	→	label=COVID-19	[9, 0, 0, 0]	77 : 8 : 8 : 8	0.769	1
11	n77≥0.28271484375 AND n16≥0.146728515625	→	label=SARS	[0, 0, 7, 0]	9 : 9 : 73 : 9	0.727	2
12	n954≥0.455078125	→	label=Pneumocystis	[0, 8, 0, 0]	8 : 75 : 8 : 8	0.750	1
13	n1480≥0.1490478515625 AND n20≥0.1337890625	→	label=COVID-19	[8, 0, 0, 0]	75 : 8 : 8 : 8	0.750	2
14	n120≥0.492919921875	→	label=Streptococcus	[0, 0, 0, 7]	9 : 9 : 9 : 73	0.727	1
15	n366≥0.267822265625	→	label=Pneumocystis	[0, 5, 0, 0]	11 : 67 : 11 : 11	0.667	1
16	n11≥0.1981201171875	→	label=SARS	[0, 0, 4, 0]	12 : 12 : 62 : 12	0.625	1
17	n2≥0.3583984375	→	label=COVID-19	[2, 0, 0, 0]	50 : 17 : 17 : 17	0.500	1
18	n0≤0.63525390625	→	label=Streptococcus	[0, 0, 0, 2]	17 : 17 : 17 : 50	0.500	1
19	TRUE	→	label=COVID-19	[118, 33, 2...	56 : 16 : 11 : 17	0.559	0

Fig. 12 CN2 Rule Induction and Evaluation measure parameter as Laplace accuracy, higher classification accuracy and a few induced easy-to-read rules

3.4 Parameters of Tree Classifier

The *Min. number of instances in leaves* parameter has no huge influence on the *Tree* toolbox's CA performance and its results. Such that, when this parameter was set to 2, its CA is 58.4% with a tree generating of 35 nodes and 18 leaves. While, when this parameter set to 3, its CA changed to 59.6% with the same tree of 35 nodes and 18 leaves, as illustrated in Fig. 13.

The interpretability rules that generate the unbalanced binary tree, as shown in Fig. 13, will be as follows:

- The parent node, *root*, generates the left-side child, *node*, when ($n21 \leq 0.283447$).
- The root generates the right-side node when ($n21 > 0.283447$).
- The left-side and right-side nodes then generate their children in the same fashion, and so on …

According to these interpretability rules, the percentile of cases was calculated based on the *nXXXX*. Note that the *n21*, for instance, is one of the 2048 features

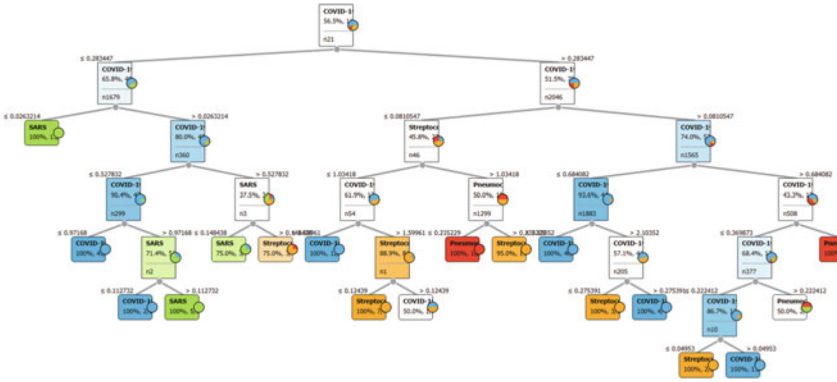


Fig. 13 Orange Tree viewer toolbox of 35 nodes and 18 leaves tree at either 2 or 3 of the Min. number of instances in leaves parameter selection

Table 2 CA based on Number of neighbors and Weight parameters

Number of neighbors	Weights	
	Uniform	Distance
5	68.5%	75.3%
10	61.8%	75.3%

that were obtained previously from the *Inception v3 Model (Training)* toolbox on the training dataset.

3.5 Parameters of kNN Classifier

The *kNN* toolbox produces different CA based on *Number of neighbors* and *Weight* parameters. Table 2 presents the generated CA based on these two parameters. Note that, there were insignificant changes in its CA for the 5 or 10 selection of the *Number of neighbors* parameter as well as for the *Uniform* or *Distance* selection of the *Weight* parameter.

3.6 Parameters of Logistic Regression Classifier

The *Logistic Regression* toolbox generates different CA depending on the *Regularization type* parameter. Therefore, when this parameter was set to *Ridge (L2)* [39], then its CA was 95.5%. While, when this parameter was set to *Lasso (L1)* [40] then its CA changed be 84.3%. Since the *Lasso (L1)* parameter shrinks the extreme values of each sample towards its central values, then this could cause the loss of important information during the classification process than the *Ridge (L2)* parameter. Note that the *Strength* parameter was set to $C = 1$.

3.7 Parameters of Neural Network Classifier

The *Neural Network* toolbox produces different CA based on its *Neurons in hidden layers*, *Activation*, and *Solver* parameters. In our work, this classifier was implemented as an MLP of 2 hidden layers, due to the MLP architecture of more hidden layers had the same generated CA with the same parameters selection. The following activation functions of *Logistic*, *tanh*, and *ReLU* [41] as well as the solver optimizers of *SGD* [42] and *Adam* [43] were selected in this work, due to their robustness and non-linearity behaviors. Note that the *Regularization* parameter was set to 0.9 and the *Maximal number of iterations* was set to 100. Table 3 states the generated CA based on these three parameters.

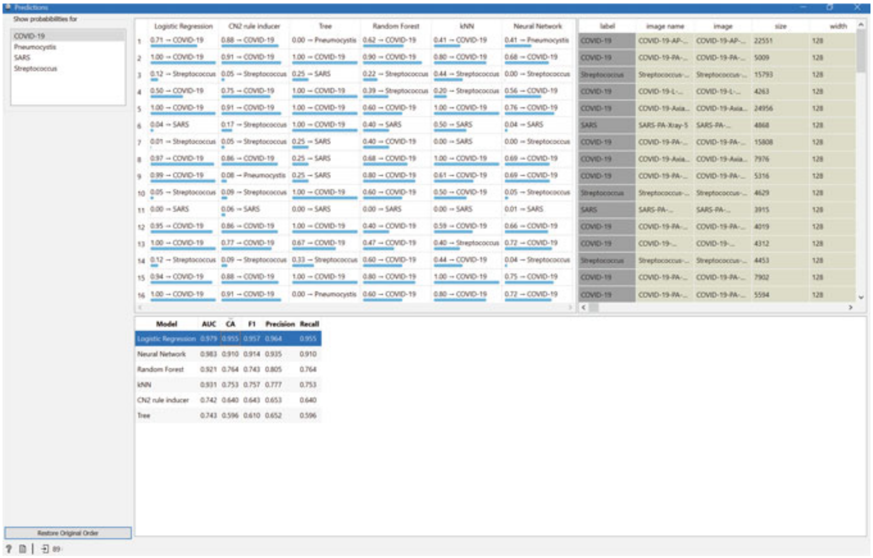
The best-chosen CA was 91.0% for the MLP architecture of the (100 × 100 × 4) neurons with ReLU activation and Adam solver parameters selection, as well as of the (500 × 500 × 4) neurons with tanh activation and Adam solver parameters selection. However, the small MLP architecture of (100 × 100 × 4) neurons was chosen due to its fewer number of the utilized neurons that decrease the classification time, rather than the medium MLP architecture of (500 × 500 × 4) neurons.

4 Results and Discussion

In our work, the obtained human-classifiable results were based on the following factors: (1) the number of medical samples in the training and test datasets, (2) the seven proposed machine learning classifiers, (3) the selected and tuned parameters for each classifier, and (4) the predicted probabilities. Moreover, the *Predictions* toolbox was implemented to visually illustrate the predicted probabilities for the test dataset based on their labels as well as the outcome signals from the classifiers based on the training dataset. These predicted probabilities are CA, AUC (Area under ROC—Receiver Operating Characteristic), Precision, Recall, and F1 (a weighted harmonic mean of Precision and Recall) that used to compute the statistical performance of a machine learning classifier [44], as demonstrated in Fig. 14. Note that, in this

Table 3 CA based on Neurons in hidden layers, Activation, and Solver parameters

Activation/solver	Neurons in hidden layers (%)		
	100 × 100 × 4	500 × 500 × 4	1000 × 1000 × 4
Logistic/SGD	57.3	57.3	57.3
Logistic/Adam	57.3	57.3	57.3
tanh/SGD	78.7	88.8	86.5
tanh/Adam	85.9	91.0	85.4
ReLU/SGD	67.4	85.4	77.5
ReLU/Adam	91.0	79.8	65.2



	Logistic Regression	CN2 rule inducer	Tree	Random Forest	kNN	Neural Network	label	image name	image	size	width
1	0.71 → COVID-19	0.88 → COVID-19	0.00 → Pneumocystis	0.62 → COVID-19	0.41 → COVID-19	0.41 → Pneumocystis	COVID-19	COVID-19-AP-...	COVID-19-AP-...	22551	128
2	1.00 → COVID-19	0.91 → COVID-19	1.00 → COVID-19	0.90 → COVID-19	0.80 → COVID-19	0.68 → COVID-19	COVID-19	COVID-19-PA-...	COVID-19-PA-...	5009	128
3	0.12 → Streptococcus	0.05 → Streptococcus	0.25 → SARS	0.22 → Streptococcus	0.44 → Streptococcus	0.00 → Streptococcus	Streptococcus	Streptococcus-...	Streptococcus-...	15793	128
4	0.50 → COVID-19	0.75 → COVID-19	1.00 → COVID-19	0.39 → Streptococcus	0.20 → Streptococcus	0.56 → COVID-19	COVID-19	COVID-19-L-...	COVID-19-L-...	4263	128
5	1.00 → COVID-19	0.91 → COVID-19	1.00 → COVID-19	0.60 → COVID-19	1.00 → COVID-19	0.76 → COVID-19	COVID-19	COVID-19-Asia...	COVID-19-Asia...	24956	128
6	0.04 → SARS	0.17 → Streptococcus	1.00 → COVID-19	0.40 → SARS	0.50 → SARS	0.04 → SARS	SARS	SARS-PA-Xray-5	SARS-PA-...	4868	128
7	0.01 → Streptococcus	0.05 → Streptococcus	0.25 → SARS	0.40 → COVID-19	0.00 → SARS	0.00 → Streptococcus	COVID-19	COVID-19-PA-...	COVID-19-PA-...	15808	128
8	0.97 → COVID-19	0.86 → COVID-19	0.25 → SARS	0.68 → COVID-19	1.00 → COVID-19	0.69 → COVID-19	COVID-19	COVID-19-Asia...	COVID-19-Asia...	7976	128
9	0.99 → COVID-19	0.08 → Pneumocystis	0.25 → SARS	0.80 → COVID-19	0.61 → COVID-19	0.69 → COVID-19	COVID-19	COVID-19-PA-...	COVID-19-PA-...	5316	128
10	0.05 → Streptococcus	0.09 → Streptococcus	1.00 → COVID-19	0.60 → COVID-19	0.50 → COVID-19	0.05 → Streptococcus	Streptococcus	Streptococcus-...	Streptococcus-...	4629	128
11	0.00 → SARS	0.06 → SARS	0.00 → SARS	0.00 → SARS	0.00 → SARS	0.01 → SARS	SARS	SARS-PA-...	SARS-PA-...	3915	128
12	0.95 → COVID-19	0.86 → COVID-19	1.00 → COVID-19	0.40 → COVID-19	0.59 → COVID-19	0.66 → COVID-19	COVID-19	COVID-19-PA-...	COVID-19-PA-...	4019	128
13	1.00 → COVID-19	0.77 → COVID-19	0.67 → COVID-19	0.47 → COVID-19	0.40 → Streptococcus	0.72 → COVID-19	COVID-19	COVID-19-...	COVID-19-...	4312	128
14	0.12 → Streptococcus	0.09 → Streptococcus	0.33 → Streptococcus	0.60 → COVID-19	0.44 → COVID-19	0.04 → Streptococcus	Streptococcus	Streptococcus-...	Streptococcus-...	4453	128
15	0.94 → COVID-19	0.88 → COVID-19	1.00 → COVID-19	0.80 → COVID-19	1.00 → COVID-19	0.75 → COVID-19	COVID-19	COVID-19-PA-...	COVID-19-PA-...	7902	128
16	1.00 → COVID-19	0.91 → COVID-19	0.00 → Pneumocystis	0.60 → COVID-19	0.80 → COVID-19	0.72 → COVID-19	COVID-19	COVID-19-PA-...	COVID-19-PA-...	5594	128

Model	AUC	CA	F1	Precision	Recall
Logistic Regression	0.979	0.955	0.957	0.964	0.955
Neural Network	0.983	0.910	0.914	0.935	0.910
Random Forest	0.921	0.764	0.743	0.805	0.764
kNN	0.931	0.753	0.757	0.777	0.753
CN2 rule inducer	0.742	0.640	0.643	0.653	0.640
Tree	0.743	0.596	0.610	0.652	0.596

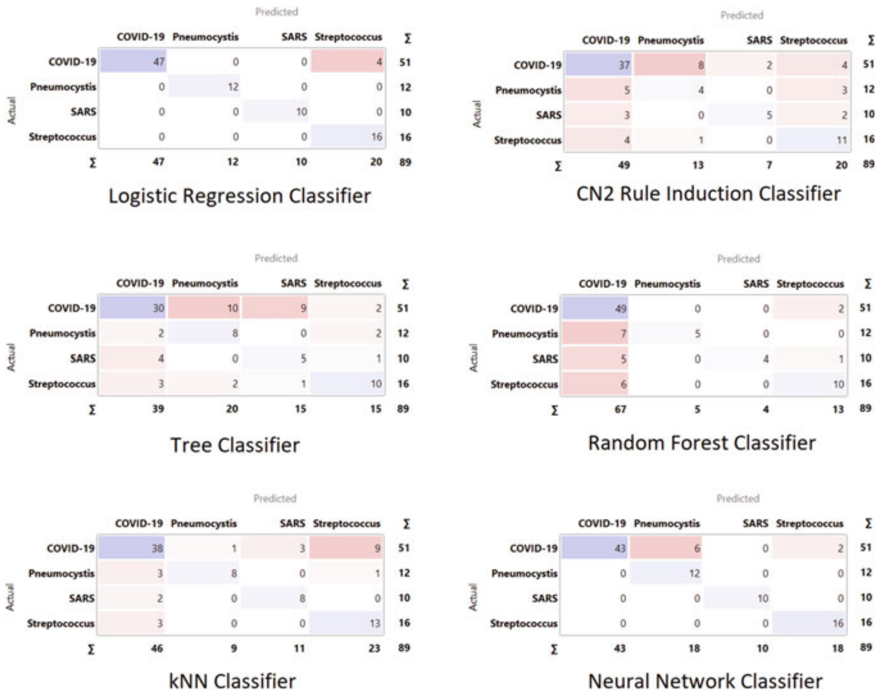
Fig. 14 Orange Predictions toolbox, (top) the blue bars show the strength of each classifier in predicting the four infected cases individually or as a group, and (bottom) the predicted probabilities (CA, AUC, Precision, Recall, and F1) of computing the statistical performance for each classifier

Predictions toolbox, the cases can be chosen individually or as a group to show the best-chosen classifier(s) that predicted them, as denoted by the lengths, or *strengths*, of the blue bars.

Moreover, other Orange toolboxes, such as *Confusion Matrix, Linear Projection*, and *Sieve Diagram*, were implemented to calculate the correlations and illustrate the projections between the training dataset and the test dataset for each machine learning classifier.

4.1 Results from Confusion Matrix

Figure 15 illustrates the correlation between the actual and the predicted cases on the samples from the test dataset as comparable matrices for each classifier regarding the four infected cases. It was observed that the COVID-19 samples were more classifiable under the Logistic Regression and Neural Network than the other classifiers, and especially for the Logistic Regression classifier since it has only four mismatched COVID-19 cases with Streptococcus cases, then followed by the Neural Network classifier that has eight mismatched COVID-19 cases with Pneumocystis and Streptococcus cases.



Logistic Regression Classifier

Actual \ Predicted	COVID-19	Pneumocystis	SARS	Streptococcus	Σ
COVID-19	47	0	0	4	51
Pneumocystis	0	12	0	0	12
SARS	0	0	10	0	10
Streptococcus	0	0	0	16	16
Σ	47	12	10	20	89

CN2 Rule Induction Classifier

Actual \ Predicted	COVID-19	Pneumocystis	SARS	Streptococcus	Σ
COVID-19	37	8	2	4	51
Pneumocystis	5	4	0	3	12
SARS	3	0	5	2	10
Streptococcus	4	1	0	11	16
Σ	49	13	7	20	89

Tree Classifier

Actual \ Predicted	COVID-19	Pneumocystis	SARS	Streptococcus	Σ
COVID-19	30	10	9	2	51
Pneumocystis	2	8	0	2	12
SARS	4	0	5	1	10
Streptococcus	3	2	1	10	16
Σ	39	20	15	15	89

Random Forest Classifier

Actual \ Predicted	COVID-19	Pneumocystis	SARS	Streptococcus	Σ
COVID-19	49	0	0	2	51
Pneumocystis	7	5	0	0	12
SARS	5	0	4	1	10
Streptococcus	6	0	0	10	16
Σ	67	5	4	13	89

kNN Classifier

Actual \ Predicted	COVID-19	Pneumocystis	SARS	Streptococcus	Σ
COVID-19	38	1	3	9	51
Pneumocystis	3	8	0	1	12
SARS	2	0	8	0	10
Streptococcus	3	0	0	13	16
Σ	46	9	11	23	89

Neural Network Classifier

Actual \ Predicted	COVID-19	Pneumocystis	SARS	Streptococcus	Σ
COVID-19	43	6	0	2	51
Pneumocystis	0	12	0	0	12
SARS	0	0	10	0	10
Streptococcus	0	0	0	16	16
Σ	43	18	10	18	89

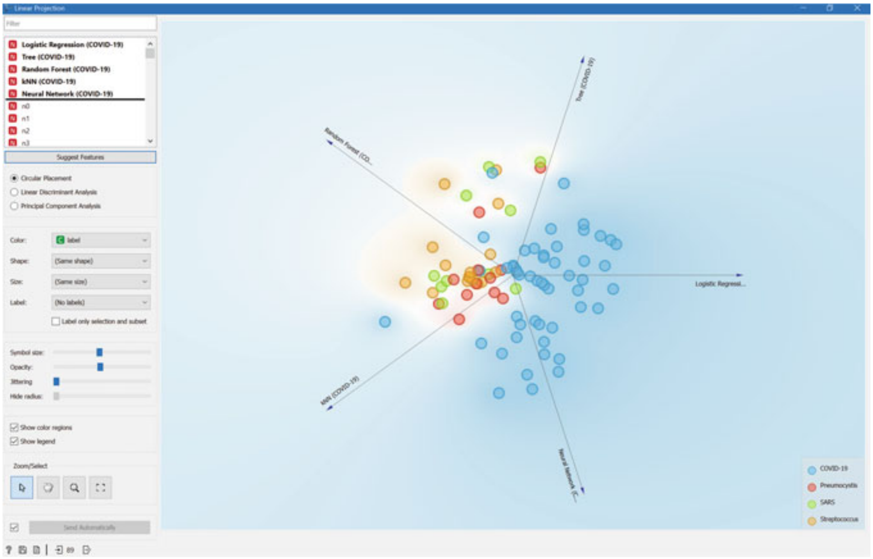
Fig. 15 Confusion matrices for each classifier, and the Logistic Regression classifier performs well on COVID-19 cases classification

4.2 Results from Linear Projection

As demonstrated in Fig. 16, the COVID-19 samples, as the blue dots, were selected to be linearly projected with the Logistic Regression, Tree, Random Forest, kNN, and Neural Network classifiers to check their accurate classification and performance. It was visually noted that the COVID-19 samples were more classifiable under the Logistic Regression and Neural Network than the other classifiers, and especially for the Logistic Regression classifier due to a large number of blue dots around its axis, then followed by the Neural Network classifier.

4.3 Results from Sieve Diagram

Orange *Sieve Diagram* toolbox provides the N samples visualization of the test dataset along with a pair of classifiers, as well as shows the *Sieve Rank* (X^2). The darker blue regions are the stronger is the relationship of a given case for a pair of classifiers. This can be influenced on the X^2 as well, so that the higher X^2 gives a better illustration of the stronger relationship for a given case among a pair of classifiers.



Linear Projection

Filter

- Logistic Regression (COVID-19)
- Tree (COVID-19)
- Random Forest (COVID-19)
- kNN (COVID-19)
- Neural Network (COVID-19)
- n0
- n1
- n2
- n3

Suggest Features

● Circular Placement
○ Linear Discriminant Analysis
○ Principal Component Analysis

Color: label
Shape: (Same shape)
Size: (Same size)
Label: (No labels)
□ Label only selection and subset

Symbol size:
Opacity:
Jittering
Hide radius:

☑ Show color regions
☑ Show legend

Zoom/Select

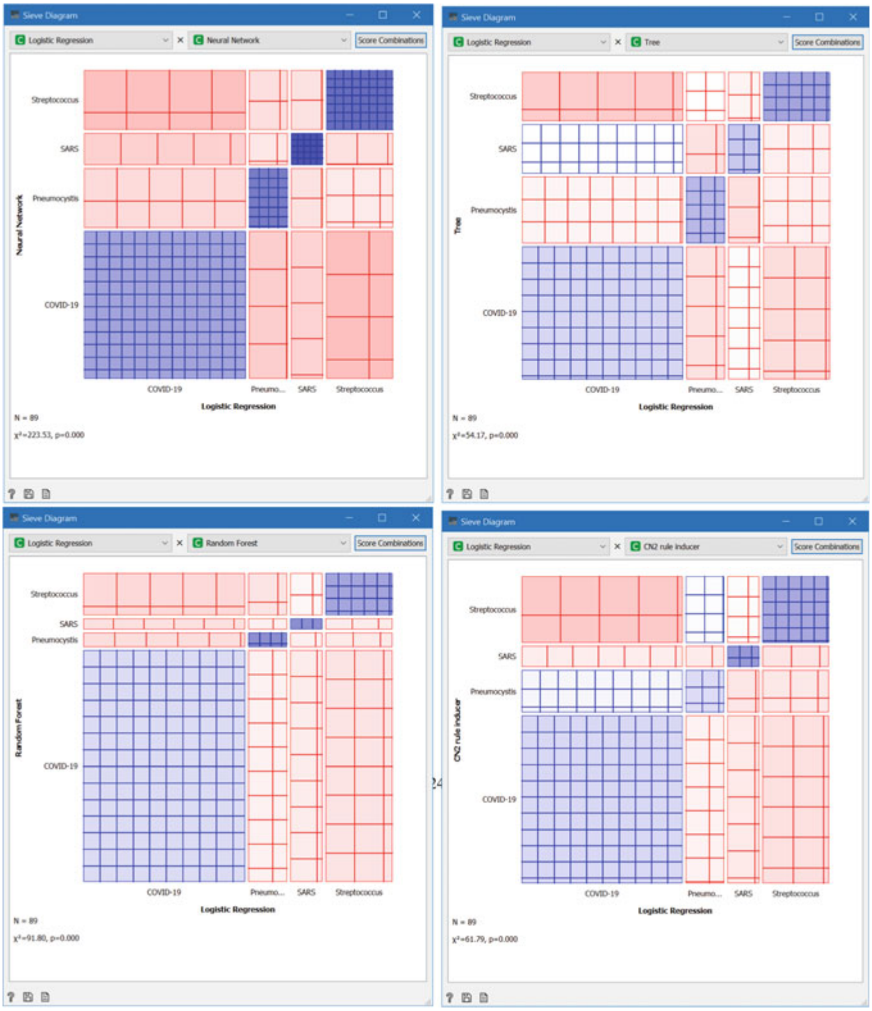
☑ Send Automatically

Fig. 16 Linear projection of COVID-19 samples, as the blue dots, within the five classifiers, and the Logistic Regression classifier performs well on COVID-19 cases classification

Figure 17 shows different Sieve diagrams for different pairs of classifiers, and it was visually observed that the COVID-19 samples were more classifiable under the pair of Logistic Regression and Neural Network classifiers. This observation was based on darker blue regions and higher X^2 than for other pairs of classifiers. On the other hand, the COVID-19 samples were less classifiable under the pairs of Logistic Regression and Tree/Random Forest/CN2 Rule Inducer classifiers, which have lighter blue regions and lower X^2.

5 Conclusions

When more medical images regarding the COVID-19 datasets have been clinically provided and publicly available, this will open the chance to do more research in this field as well as increase the number of samples to the classifiers, which yields to get better modeling performance and classification accuracy. Hence, larger datasets of various diseases' cases will contribute to do more labeling assignments, however less data augmentation would be required. The image scaling of 128 × 128 pixels was chosen, in this work, for the purpose of fast processing time, but this downscaling may also affect the hidden features of these diagnostic images, which gives at the end no better features extraction and detection using the Orange Inception v3 model toolbox. Our work was implemented using Orange software, a visual analytical tool for epidemiology specialists that have little knowledge in machine learning


Sieve Diagram

Logistic Regression × Neural Network Score Combinations

N = 89

χ^2=223.53, p=0.000

Sieve Diagram

Logistic Regression × Tree Score Combinations

N = 89

χ^2=54.17, p=0.000

Sieve Diagram

Logistic Regression × Random Forest Score Combinations

N = 89

χ^2=91.80, p=0.000

Sieve Diagram

Logistic Regression × CN2 rule inducer Score Combinations

N = 89

χ^2=61.79, p=0.000

Fig. 17 Sieve diagrams for pairs of classifiers, N is the number of test samples and X^2 is Sieve rank. (top-left) the pair of Logistic Regression and Neural Network classifiers performs well on COVID-19 cases classification with darker blue regions and higher X^2

techniques and limited programming skills, to allow them to focus on COVID-19 cases analysis and classification with ease of use and simplicity of understanding and manipulating.

Some of the toolboxes' parameters do not affect the overall workflow, while others do. Table 4 describes the parameters that have been carefully chosen for better rules, easy visualization, as well as higher performance and CA.

Based on Table 4, the *Predictions* toolbox demonstrates different CA for different classifiers. The higher CA is, the better classifier does, as categorized by Table 5.

Table 4 Parameters selection for higher performance and CA

Toolbox	Parameter(s)	Setting
Distances	Distance metric	Cosine
CN2 Rule induction	Evaluation measure	Laplace accuracy
Tree	Min. number of instances in leaves	3
Random forest	Number of trees	10
kNN	Number of neighbors	5
	Weight	Distance
Logistic regression	Regularization type	Ridge (L2)
Neural network	Neurons in hidden layers	100 × 100 × 4
	Activation	ReLU
	Solver	Adam

Table 5 CA ranking for each classifier

Classifier	CA (%)
Logistic regression	95.5
Neural network	91.0
Random forest	76.4
kNN	75.3
CN2 rule induction	64.0
Tree	59.6

Based on the *Confusion Matrix* toolbox, as in Fig. 15 shown previously, the Logistic Regression classifier has a better matching between the actual and the predicted cases for the samples from the test dataset, and then followed by the Neural Network classifier. While the Tree classifier has the worst matching pattern. Therefore, this is in agreement with Table 5. Based on the *Linear Projection* toolbox, as in Fig. 16 shown previously, the Logistic Regression and Neural Network classifiers have the most COVID-19 cases projection along their axes than the other classifiers. Therefore, this is in agreement with Table 5.

Based on the *Sieve Diagram* toolbox, as in Fig. 17 shown previously, the pair of Logistic Regression and Neural Network classifiers have the most test samples frequencies of COVID-19 cases, due to the darker blue regions and higher X^2. Therefore, this is in agreement with Table 5. Based on the careful parameters selection and tuning for each classifier, the *Predictions* toolbox, the *Confusion Matrix* toolbox, the *Linear Projection* toolbox, the *Sieve Diagram* toolbox, and the CA from Table 5, the Logistic Regression model was the appropriate best-chosen classifier in identifying and classifying the COVID-19 cases among the other similar human-being lungs infected cases (SARS, Pneumocystis, and Streptococcus). The achieved CA for the Logistic Regression classifier was up to 95.5%. Figure 18 demonstrates the completed

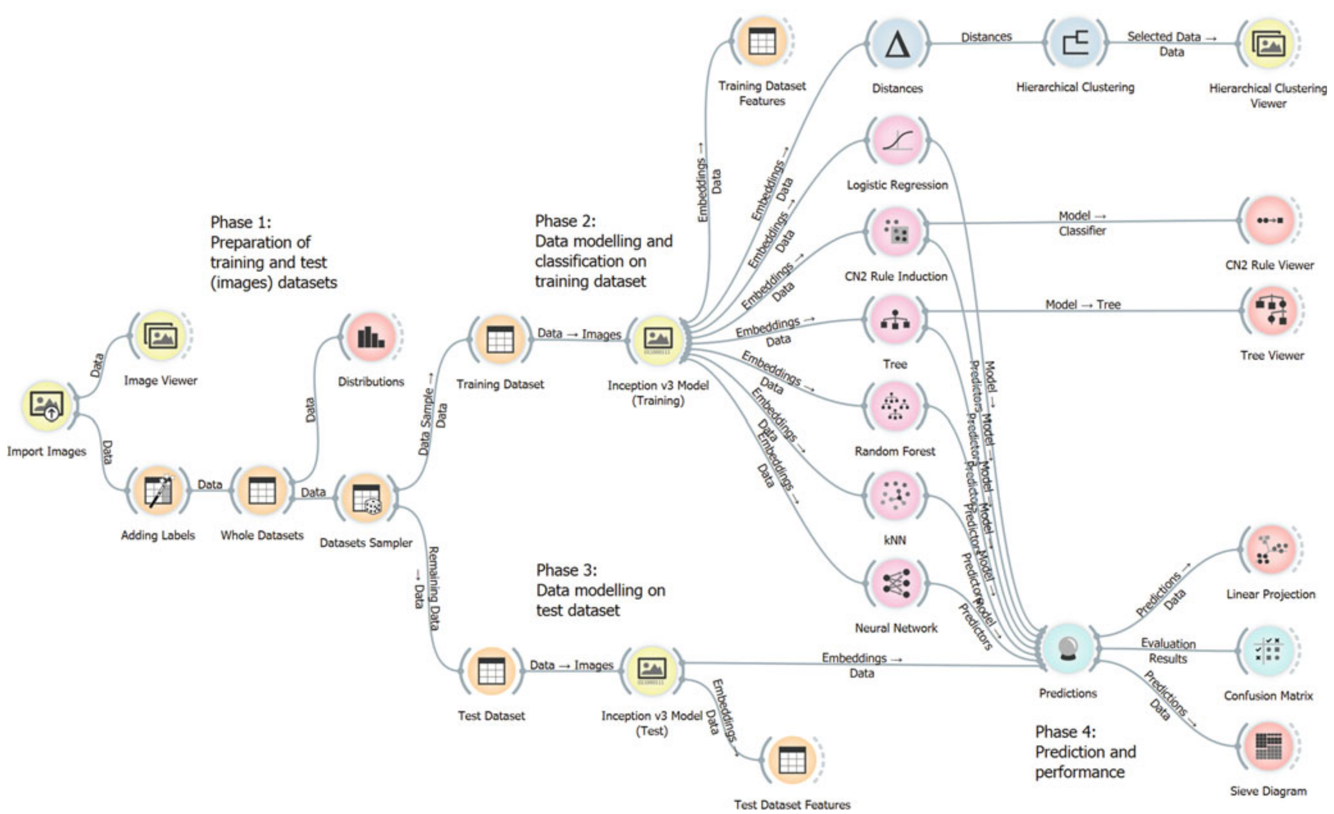


Fig. 18 Orange phases, toolboxes, and signals workflow layout

layout of all connected phases, toolboxes, and signals to achieve the overall Orange workflow for COVID-19 features detection and extraction using machine learning classifiers.

References

1. World Health Organization. Coronavirus disease (COVID-19) pandemic. WHO.int. https://www.who.int/emergencies/diseases/novel-coronavirus-2019. Accessed 4 May 2021
2. Marmanis, D., Datcu, M., Esch, T., Stilla, U.: Deep learning earth observation classification using ImageNet pretrained networks. IEEE Geosci. Remote Sens. Lett. **13**(1), 105–109 (2015)
3. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. Adv. Neural. Inf. Process. Syst. **25**, 1097–1105 (2012)
4. Rajpurkar, P., Irvin, J., Zhu, K., Yang, B., Mehta, H., Duan, T., Ding, D., Bagul, A., Langlotz, C., Shpanskaya, K., Lungren, M.P.: CheXNet: Radiologist-level pneumonia detection on chest X-rays with deep learning (2017). arXiv preprint arXiv:1711.05225
5. Ayan, E., Ünver, H.M.: Diagnosis of pneumonia from chest X-ray images using deep learning. In: 2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT) pp. 1–5
6. Stephen, O., Sain, M., Maduh, U.J., Jeong, D.U.: An efficient deep learning approach to Pneumonia classification in healthcare. J. Healthc. Eng. (2019). https://doi.org/10.1155/4180949
7. Alimadadi, A., Aryal, S., Manandhar, I., Munroe, P.B., Joe, B., Cheng, X.: Artificial intelligence and machine learning to fight COVID-19. Physiol. Genomics **52**, 200–202 (2020). https://doi.org/10.1152/00029.2020
8. Pinter, G., Felde, I., Mosavi, A., Ghamisi, P., Gloaguen, R.: COVID-19 pandemic prediction for Hungary; a hybrid machine learning approach. Mathematics **8**(6) (2020). https://doi.org/10.3390/math8060890
9. Elaziz, M.A., Hosny, K.M., Salah, A., Darwish, M.M., Lu, S., Sahlol, A.T.: New machine learning method for image-based diagnosis of COVID-19. PLOS ONE **15**(6) (2020). https://doi.org/10.1371/journal.pone.0235187
10. Sujath, R., Chatterjee, J.M., Hassanien, A.E.: A machine learning forecasting model for COVID-19 pandemic in India. Stoch. Env. Res. Risk Assess. **34**, 959–972 (2020). https://doi.org/10.1007/s00477-020-01827-8
11. Ardabili, S.F., Mosavi, A., Ghamisi, P., Ferdinand, F., Varkonyi-Koczy, A.R., Reuter, U., Rabczuk, T., Atkinson, P.M.: COVID-19 outbreak prediction with machine learning. Algorithms **13**(10) (2020). https://doi.org/10.3390/a13100249
12. Brinati, D., Campagner, A., Ferrari, D., Locatelli, M., Banfi, G., Cabitza, F.: Detection of COVID-19 infection from routine blood exams with machine learning: a feasibility study. J. Med. Syst. **44**(135), 1–12 (2020). https://doi.org/10.1007/s10916-020-01597-4
13. Cheng, F.Y., Joshi, H., Tandon, P., Freeman, R., Reich, D.L., Mazumdar, M., Kohli-Seth, R., Levin, M., Timsina, P., Kia, A.: Using machine learning to predict ICU transfer in hospitalized COVID-19 patients. J. Clin. Med. **9**(6) (2020). https://doi.org/10.3390/jcm9061668
14. Rustam, F., Reshi, A.A., Mehmood, A., Ullah, S., On, B.W., Aslam, W., Choi, G.S.: COVID-19 future forecasting using supervised machine learning models. IEEE Access **8**, 101489–101499 (2020). https://doi.org/10.1109/ACCESS.2020.2997311
15. Kumar, S.R.: Novel Corona Virus 2019 Dataset V151. Distributed by Kaggle Inc. https://www.kaggle.com/sudalairajkumar/novel-corona-virus-2019-dataset
16. Allen Institute for AI: COVID-19 Open Research Dataset Challenge (CORD-19) V92. Distributed by Kaggle Inc. https://www.kaggle.com/allen-institute-for-ai/CORD-19-research-challenge

17. Orange: University of Ljubljana (2021). Accessed Apr 2021. https://orangedatamining.com
18. Gärtner, T., Lloyd, J.W., Flach, P.A.: Kernels and distances for structured data. Mach. Learn. **57**(3), 205–232 (2004)
19. Collins, M., Schapire, R.E., Singer, Y.: Logistic regression, AdaBoost and Bregman distances. Mach. Learn. **48**(1), 253–285 (2002)
20. Dreiseitl, S., Ohno-Machado, L.: Logistic regression and artificial neural network classification models: a methodology review. J. Biomed. Inform. **35**, 352–359 (2002)
21. Clark, P., Boswell, R.: Rule induction with CN2: some recent improvements. In: Machine Learning—Proceedings of the Fifth European Conference (EWSL-91), pp. 151–163 (1991)
22. Džeroski, S., Grbovic, J., Walley, W.J., Kompare, B.: Using machine learning techniques in the construction of models. II. Data analysis with rule induction. Ecol. Model. **95**(1), 95–111 (1997)
23. Dietterich, T.G., Kong, E.B.: Machine learning bias, statistical bias, and statistical variance of decision tree algorithms pp. 0–13. Technical report, Dept. of Computer Science, Oregon State University, USA (1995)
24. Olson, R.S., Moore, J.H.: TPOT: A tree-based pipeline optimization tool for automating machine learning. In: Workshop on Automatic Machine Learning (ICML), pp. 66–74 (2016)
25. Segal M.R.: Machine learning benchmarks and random forest regression. UCSF: Center for Bioinformatics and Molecular Biostatistics (2004). Retrieved from https://escholarship.org/uc/item/35x3v9t4
26. Rodriguez-Galiano, V., Sanchez-Castillo, M., Chica-Olmo, M., Chica-Rivas, M.: Machine learning predictive models for mineral prospectivity: an evaluation of neural networks, random forest, regression trees and support vector machines. Ore Geol. Rev. **71**, 804–818 (2015). https://doi.org/10.1016/j.oregeorev.2015.01.001
27. Kramer, O.: K-nearest neighbors. In: Dimensionality Reduction with Unsupervised Nearest Neighbors. Intell. Syst. Ref. Libr. **51**, 13–23 (2013). https://doi.org/10.1007/978-3-642-38652-7_2
28. Zhang, Z.: Introduction to machine learning: k-nearest neighbors. Ann. Transl. Med. **4**(11) (2016). https://doi.org/10.21037/atm.2016.03.37
29. Chen, H.: Machine learning for information retrieval: Neural networks, symbolic learning, and genetic algorithms. J. Am. Soc. Inf. Sci. **46**(3), 194–216 (1995)
30. Lampignano, J.P., Kendrick, L.E.: Bontrager's Handbook of Radiographic Positioning and Techniques, 9th edn. Mosby, USA (2017)
31. Herring, W.: Learning Radiology: Recognizing the Basics, 4th edn. Elsevier, USA (2019)
32. Abu-Mostafa, Y.S., Magdon-Ismail, M., Lin, H.T.: Learning From Data: A Short Course. AMLBook, USA (2012)
33. ACDSee: ACD Systems International Inc. (2020). https://www.acdsee.com
34. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2818–2826
35. Deng, X., Liu, Q., Deng, Y., Mahadevan, S.: An improved method to construct basic probability assignment based on the confusion matrix for classification problem. Inf. Sci. **340**, 250–261 (2016). https://doi.org/10.1016/j.ins.2016.01.033
36. Hidayatullah, R.S., Cholifah, W.N., Ambarsari, E.W., Kustian, N., Julaeha, S.: Sieve diagram for data exploration of Instagram usage habit obtained from Indonesia questioner's sample. J. Phys. **1783**(1) (2021). https://doi.org/10.1088/1742-6596/1783/1/012028
37. Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J.: Eigenfaces vs. fisherfaces: recognition using class specific linear projection. IEEE Trans. Pattern Anal. Mach. Intell. **19**(7), 711–720 (1997). https://doi.org/10.1109/34.598228
38. MacKay, D.: Information Theory, Inference and Learning Algorithms, 1st edn. Cambridge University Press, UK (2003)
39. Martínez-Martínez, J.M., Escandell-Montero, P., Soria-Olivas, E., Martín-Guerrero, J.D., Magdalena-Benedito, R., GóMez-Sanchis, J.: Regularized extreme learning machine for regression problems. Neurocomputing **74**(17), 3716–3721 (2011). https://doi.org/10.1016/j.neucom.2011.06.013

40. Tibshirani, R.: Regression shrinkage and selection via the lasso. J. Roy. Stat. Soc. **58**(1), 267–288 (1996)
41. Zhang, H., Weng, T.W., Chen, P.Y., Hsieh, C.J., Daniel, L.: Efficient neural network robustness certification with general activation functions (2018). arXiv preprint arXiv:1811.00866
42. Qian, N.: On the momentum term in gradient descent learning algorithms. Neural Netw. **12**(1), 145–151 (1999). https://doi.org/10.1016/S0893-6080(98)00116-6
43. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization (2017). arXiv preprint arXiv:1412.6980
44. Powers, D.M.W.: Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. 2020, arXiv preprint arXiv:2010.16061