Multi-objective automatic analysis of lung ultrasound data from COVID-19 patients by means of deep learning and decision trees

Leonardo Lucio Custode[a], Federico Mento[a], Francesco Tursi[b], Andrea Smargiassi[c], Riccardo Inchingolo[c], Tiziano Perrone[d,e], Libertario Demi[a], Giovanni Iacca[a]

[a] *Dept. of Information Engineering and Computer Science, University of Trento, Italy*
[b] *UOS Pneumologia di Codogno, ASST Lodi, Lodi, Italy*
[c] *Dept. of Medical and Surgical Sciences, Fondazione Policlinico Universitario A. Gemelli IRCCS, Rome, Italy*
[d] *Dept. of Internal Medicine, IRCCS San Matteo, Pavia, Italy*
[e] *Emergency Dept., Humanitas Gavazzeni, Bergamo, Italy*

Abstract

COVID-19 raised the need for automatic medical diagnosis, to increase the physicians' efficiency in managing the pandemic. Among all the techniques for evaluating the status of the lungs of a patient with COVID-19, lung ultrasound (LUS) offers several advantages: portability, cost-effectiveness, safety. Several works approached the automatic detection of LUS imaging patterns related COVID-19 by using deep neural networks (DNNs). However, the decision processes based on DNNs are not fully explainable, which generally results in a lack of trust from physicians. This, in turn, slows down the adoption of such systems. In this work, we use two previously built DNNs as feature extractors at the frame level, and automatically synthesize, by means of an evolutionary algorithm, a decision tree (DT) that aggregates in an interpretable way the predictions made by the DNNs, returning the severity of the patients' conditions according to a LUS score of prognostic value. Our results show that our approach performs comparably or better than previously reported aggregation techniques based on an empiric combination of frame-level predictions made by DNNs. Furthermore, when we analyze the evolved DTs, we discover properties about the DNNs used as feature extractors. We make our data publicly available for further development and reproducibility.

Keywords: COVID-19, lung ultrasound, decision trees, grammatical evolution, evolutionary algorithms, neuro-symbolic artificial intelligence.

*Corresponding author
Email address: giovanni.iacca@unitn.it (Giovanni Iacca)

1. Introduction

Since the outbreak of the coronavirus disease 2019 (COVID-19) pandemic, the use of lung ultrasound (LUS) has been globally and fastly spreading. Indeed, the main advantages of LUS (portability, cost-effectiveness, real-time imaging, and safety) compared to other imaging technologies such as, e.g., Computed Tomography (CT), allowed LUS to be widely adopted to evaluate the state of lungs in patients affected by COVID-19 [1–9]. Moreover, LUS can be nowadays used for patients' monitoring and for the triage of symptomatic patients [1]. In particular, LUS is often exploited to detect COVID-19 associated interstitial pneumonia and follow its evolution [2, 10]. To perform this task, different imaging protocols have been proposed together with semi-quantitative scoring systems [11]. Indeed, even though quantitative approaches aiming at assessing the condition of lung parenchyma with ultrasound are emerging [10, 12–16], these strategies are not available for emergency contexts, due to their current preliminary state. Therefore, semi-quantitative scoring systems based on specific LUS imaging patterns (e.g., vertical and horizontal artifacts, or consolidations) have been extensively exploited during the pandemic [2].

Even though a LUS quantitative analysis cannot be performed with the currently available technologies, the use of artificial intelligence (AI) for the classification of LUS frames according to a semi-quantitative scoring system can be exploited to reduce subjectivity in the evaluation and to reduce the time required to perform the analysis [17–20].

In the hereby study we exploit a standardized imaging protocol based on 14 scanning areas and on a four-level scoring system, which allows the grading of the state of lungs [2]. A recent study demonstrated how this standardized protocol and scoring system have a prognostic value when evaluating the cumulative score (sum of scores obtained in the 14 scanning areas) at exam-level [21]. We acquire 1808 LUS videos from 100 COVID-19 positive patients, which consist of 366,301 frames in total. These frames are then fed to two DNNs [17] that were previously trained to perform automatic scoring and segmentation of LUS frames according to the above-mentioned four-level scoring system [2]. We successively use the scores given as output by two DNNs (respectively for segmentation and labeling) [17] to train and test a novel automatic approach, based on decision trees (DTs) automatically synthesized by evolutionary computation, aiming at passing from frame-based labeling to video-based labeling. Specifically, we compare the video-level scores given by our automatic approach with scores given by expert clinicians. Indeed, to perform their evaluation, clinicians associate a score to each video rather than to each frame. We then assess the performance of our aggregation approach (both at video-level and exam-level) by comparing the results obtained by the proposed method with the empirical aggregation technique previously reported in [22], which represents the current state-of-the-art. We hypothesize that, even though this existing technique achieves good performance, the fact that its decisions are obtained by aggregating the outputs of the DNNs by means of a simple threshold-based approach may be sub-optimal. To overcome this limitation, we instead use a fully data-driven DT-based approach, that is in principle more flexible and does not require empirical choices of thresholds. To summarize, the main contributions of this work are the following:

1. we propose a neuro-symbolic approach to the automatic scoring of COVID-19 patients by combining DNNs and interpretable DTs;
2. we compare single-objective and multi-objective evolutionary approaches to synthesize DTs optimized w.r.t. three different metrics of interest;
3. we interpret the evolved DTs to understand their decision policies;
4. we obtain decision support systems that have both higher prognostic agreement and less variance w.r.t. the approach previously proposed in the state-of-the-art work in the field [22].

The paper is organized as follows. Firstly, we present the dataset and the design of our study, as well as the proposed method aiming at aggregating LUS frame-based predictions to obtain video-level predictions (Section 2). Successively, the results are presented (Section 3), followed by a detailed analysis of the evolved DTs (Section 4). Finally, the conclusions are derived and discussed (Section 5).

2. Materials and methods

We use the two models from [17] as feature extractors, whose outputs are aggregated and given in input to an evolved DT, which will then make a prediction of the score related to the video. A block diagram of the process is shown in Figure 1.

2.1. Data

The investigated population consists of 100 patients diagnosed as COVID-19 positive by a reverse transcription polymerase chain reaction (RT-PCR) swab test. Of the 100 patients, 63 (35 male,

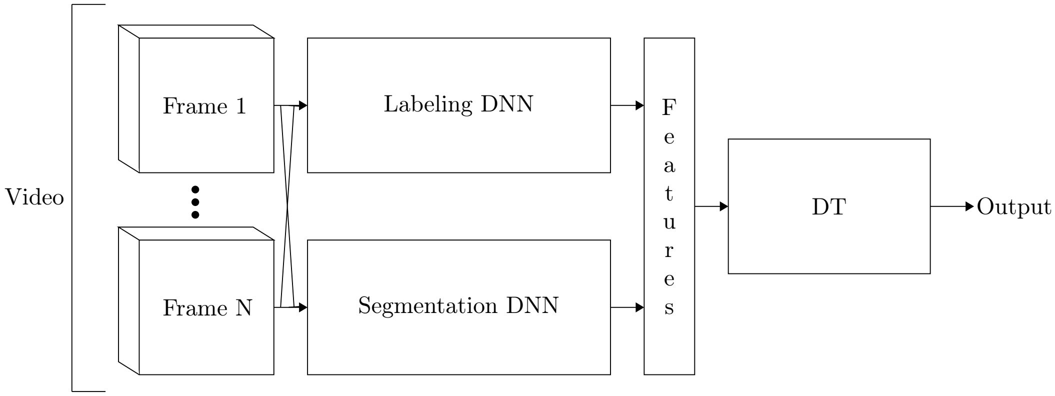


Figure 1: Block diagram of the prediction process.

28 female; ages ranging from 26 to 92 years, and average age equal to 63.72 years) were examined within the Fondazione Policlinico San Matteo (Pavia, Italy), 19 (16 male, 3 female; ages ranging from 34 to 84 years, and average age equal to 63.95 years) within the Lodi General Hospital (Lodi, Italy), and 18 (8 male, 10 female; ages ranging from 23 to 95 years, and average age equal to 52.11 years) within the Fondazione Policlinico Universitario Agostino Gemelli (Rome, Italy). As a subgroup of patients was examined multiple times, on different dates, a total of 133 LUS exams were performed (94 at Pavia, 20 at Lodi, and 19 at Rome). A total of 1808 LUS videos were thus acquired (1,290 at Pavia, 276 at Lodi, 242 at Rome), which consist of 366,301 frames (292,943 at Pavia, 44,288 at Lodi, 29,070 at Rome).

The data from Pavia have been acquired using a convex probe with an Esaote MyLab Twice scanner, and an Esaote MyLab 50, setting an imaging depth from 8 to 12 cm (depending on the patient) and an imaging frequency from 5.0 to 6.6 MHz (depending on the scanner). The data from Lodi have been acquired using a convex probe with an Esaote Mylab Sigma scanner, and a MindRay TE7, setting an imaging depth from 8 to 12 cm (depending on the patient) and an imaging frequency from 3.5 to 5.5 MHz. The data from Rome have been acquired using a convex probe with an Esaote MyLab 50, an Esaote MyLab Alpha, and a Philips IU22, setting an imaging depth from 8 to 12 cm (depending on the patient), and an imaging frequency from 3.5 to 6.6 MHz (depending on the scanner).

This study was part of a protocol that has been registered (NCT04322487) and received approval from the Ethical Committee of the Fondazione Policlinico Universitario San Matteo (protocol 20200063198), of Milano area 1, the Azienda Socio-Sanitaria Territoriale Fatebenefratelli-Sacco (protocol N0031981), of the Fondazione Policlinico Universitario Agostino Gemelli, Istituto di Ricovero e Cura a Carattere Scientifico (protocol 0015884/20 ID 3117). All patients gave informed consent.

The patients were examined by applying a standardized acquisition protocol based on 14 scanning areas [2]. This protocol is based on a four-level scoring system consisting in assigning a score that rages from 0 to 3, depending on the observed LUS patterns, with score 0 indicating a healthy lung surface, and 1, 2, 3 an increasingly altered lung surface [2]. All the 1808 LUS videos were thus scored by LUS medical experts (in this case, the authors T.P., F.T., and A.S.). Each expert labeled the videos acquired by himself, i.e., T.P. labeled videos from Pavia, F.T. from Lodi, and A.S. from Rome. The distribution of scores assigned at video-level by the experts is shown in Figure 2.

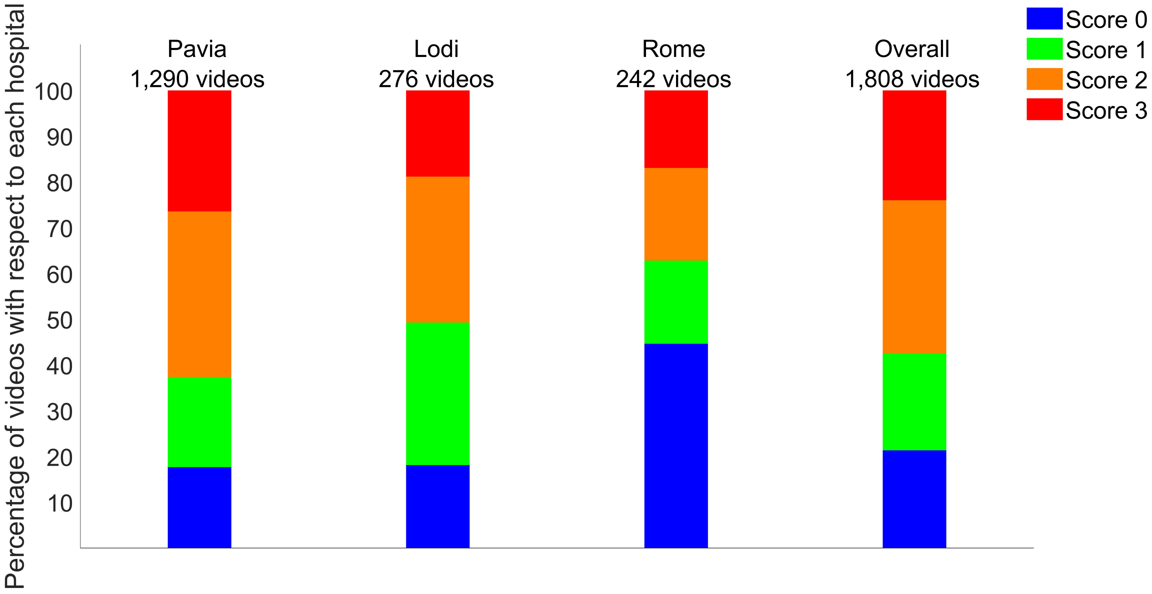


Figure 2: The distribution of scores assigned at video-level by the three clinical experts is shown. The percentage of scores 0, 1, 2, and 3 is shown for each hospital (Pavia, Lodi, and Rome) and for the entire dataset (overall). The total number of videos for each group is provided on top.

2.1.1. Inputs

All the 1808 videos are fed to the two DNNs presented by Roy *et al.* [17], i.e., a labeling DNN derived from Spatial Transformer Networks and a segmentation DNN derived from U-Nets and DeepLab v3+. The former provides as output a score for each input frame, whereas the latter provided semantic segmentation and assigned one or multiple scores to each frame [17]. As the

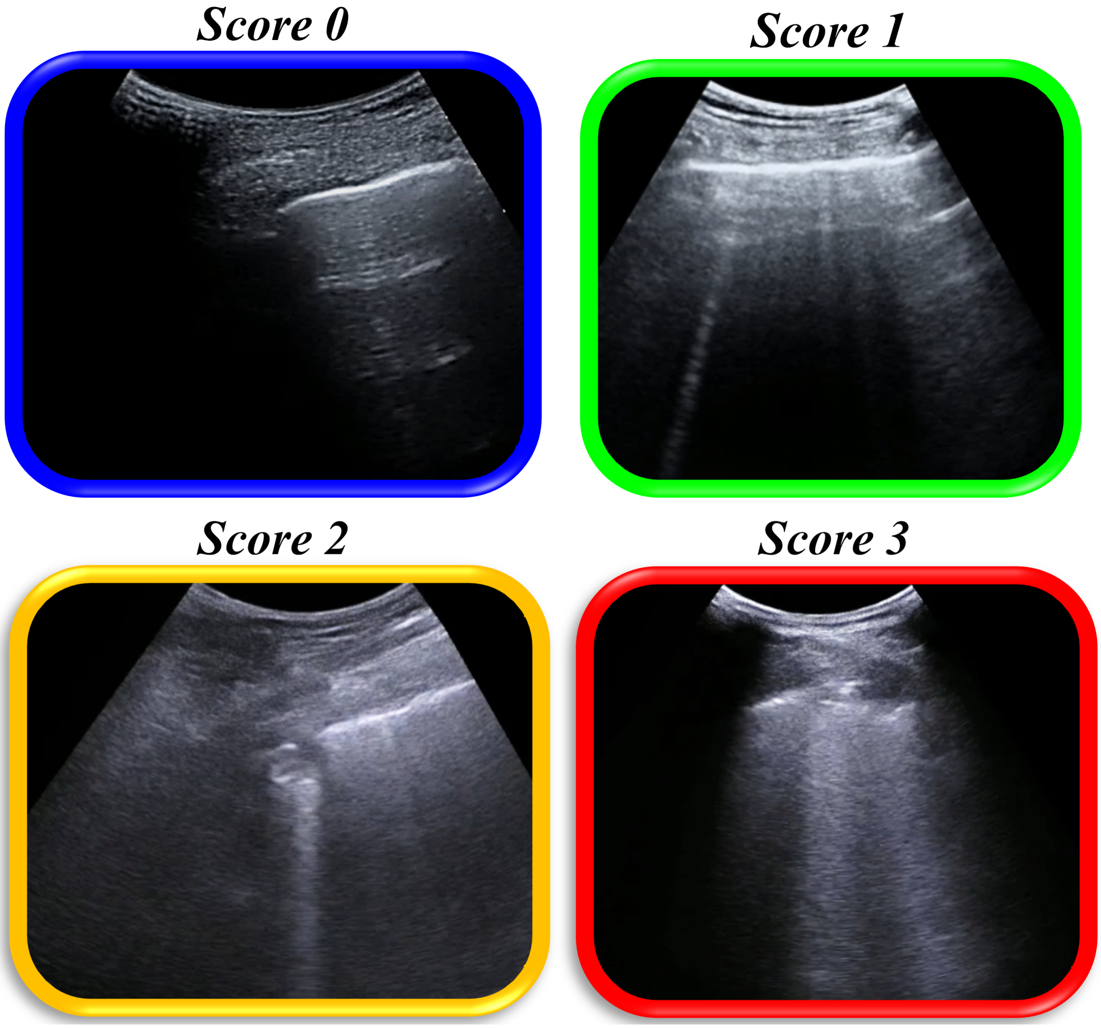


Figure 3: Examples of frames labeled as scores 0, 1, 2, and 3 are shown.

segmentation DNN can provide multiple scores for the same frame, we assign only the highest score predicted by this DNN to each considered frame (i.e., the worst-case score). Moreover, it is important to highlight how the segmentation DNN could provide no scores in output (if it does not find any relevant LUS pattern). Therefore, an extra score indicating the absence of LUS patterns (characteristic of the four scores) is considered when evaluating the output provided by segmentation DNN (we called it score -1). These two DNNs have been previously trained with the dataset presented by Roy *et al.* [17], which does not depend on the dataset exploited in the hereby

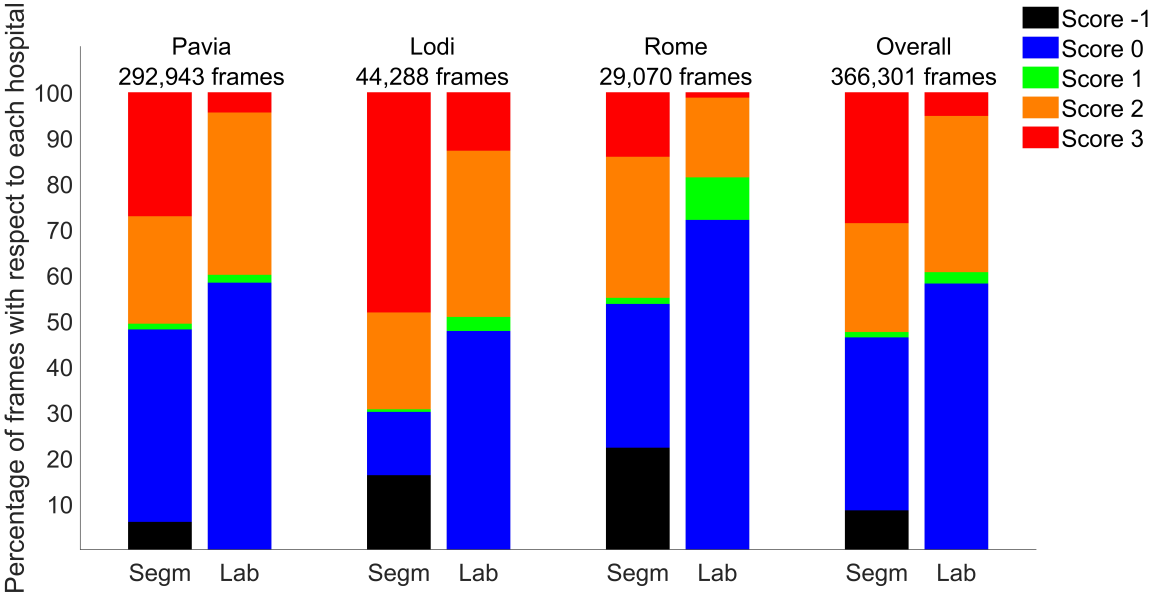


Figure 4: The distribution of scores assigned at frame-level by the labeling and segmentation DNNs presented in [17]), exploited in this work, is shown. The percentage of scores 0, 1, 2, and 3 is shown for each hospital (Pavia, Lodi, and Rome) and for the entire dataset (overall). The frame-level scores given by each architecture are shown separately. As the segmentation DNN can provide multiple scores for the same frame, we scored each frame with the worst-case (i.e., maximum) score predicted by the segmentation DNN . As the segmentation DNN could provide no output (if it does not find any relevant LUS pattern), an extra distribution represented as score -1 is observable in the left bars. The total number of frames for each group is provided on top.

work. Figure 3 shows examples of frames labeled as scores 0, 1, 2, and 3. Figure 4 shows the distribution of scores assigned at frame-level by the exploited two DNNs. Considering the entire dataset (see the overall distribution, Figure 4, right), there is a high percentage of score 0 and 2 for both labeling and segmentation DNNs, whereas score 1 is less frequently given as output. It is also observable how the percentage of score 3 is significantly higher when looking at the segmentation DNN.

2.1.2. Targets

Given the frame-level labeling provided by the two DNNs, our target consists in finding an aggregation technique that allows us to pass from a frame-level score to a video-level score, which is the output needed by physicians to perform their clinical evaluation. Therefore, the goal of the proposed technique is to optimize three metrics of interest that compare the video-level scores obtained by our algorithm and the ones assigned by clinical experts (see Figure 2). These three metrics are: the video-level agreement, the exam-level agreement, and the prognostic-level agreement [22].

The video-level agreement consists of the percentage of videos that are correctly classified by the algorithm (i.e., the score assigned by the expert coincides with the score assigned by algorithm) [22]. We also evaluate the video-level agreement when allowing a disagreement up to 1 point (e.g., if the algorithm classified a video as score 2 and the expert as score 1 or 3, the evaluation is correct) [22]. To distinguish these two video-level agreements, we denote the agreement characterized by exact match between video-level scores as video-level agreement with threshold (Th) equal to 0, whereas we refer to the video-level agreement allowing a disagreement up to 1 point as video-level agreement with Th equal to 1.

The exam-level agreement is instead computed by considering the cumulative score obtained by summing the video-level scores assigned to each of the 14 scanning areas [2, 22]. Specifically, we compute the exam-level agreement as the percentage of LUS exams (133 in total) having a cumulative score (ranging from 0 to $3 \times 14 = 42$) allowing a disagreement between algorithm and clinical experts of up to 2, 5, and 10 points (i.e., Th equal to 2, 5, and 10, respectively) [22]. To support the stratification between patients at high risk of clinical worsening and patients at low risk, we need to consider the prognostic value of the aforementioned protocol [2], which has been recently proven in a single-center study on 52 patients [21]. In particular, the patient is at low risk of clinical worsening when the exam-based cumulative score is less than or equal to 24, whereas the patient is at high risk of worsening when the exam-based cumulative score is greater than 24 [21].

We thus evaluate the algorithm capability of automatically stratifying these two categories of patients by measuring the prognostic-level agreement [22]. Specifically, clinical experts and algorithm are considered in prognostic agreement when both cumulative scores are less than or equal to 24 (low risk) or greater than 24 (high risk) (Th equal to 24) [22].

As we will show in detail in Section 2.3.1, it should be noted that we use a proxy for the first two metrics, i.e., the Mean Square Error (MSE), which gives us an advantage over optimizing directly the agreement. In fact, in [22] the authors use several tolerances for the video-level and exam-level agreement: this implies that in practical scenarios one should either use n objectives for each metric, where n is the number of tolerances (e.g., maximize the exam-level agreement with tolerances 2, 5, and 10); or, one should choose one tolerance among all the tolerances, which may lead to DTs that perform well only for that particular tolerance. Instead, using the MSE as we do here allows us to maximize simultaneously the agreement for all the tolerances.

2.1.3. Splitting of the data

We split the data randomly in 5 folds. To prevent data leakage, we make sure that all the data belonging to a patient are assigned to the same fold. Moreover, we use 4 folds for the training phase (i.e., to compute the fitness of the individuals) and the remaining one to assess the generalization capabilities of the best evolved DTs (i.e., as test set).

2.2. Feature extraction and aggregation of the outputs

The DT (as shown in Figure 1) expects as input video-level features. Instead, the two DNNs' input consists in single frames of each video. To convert the features from frame-level to video-level, we aggregate the outputs of each DNN. For the labeling DNN, we simply use as features the relative frequency of the prediction of each class (i.e., the argmax of the output vector of each frame). For the segmentation DNN, instead, we aggregate the features by computing the relative frequency of the worst-case (i.e., maximum) predicted classes inside each frame. This distinction between the two DNNs is needed since the segmentation DNN does not produce a single prediction but, instead, it produces a mask for the frame taken in input. Moreover, we add to the feature vector also the minimum and the maximum prediction made by the two DNNs.

The resulting feature vector is thus composed by 12 features: l_0, l_1, l_2, l_3, l_{min}, l_{max}, s_0, s_1, s_2, s_3, s_{min}, s_{max} where: l_i, $i = 0, 1, 2, 3$, represents the relative frequency of the prediction of the class i made by the labeling DNN; l_{min} and l_{max} represent the minimum and the maximum gravity level predicted by the labeling DNN; s_i, $i = 0, 1, 2, 3$, represents the number of cases in which the class i corresponds to the worst-case in the predictions made by the segmentation DNN; s_{min} and s_{max} represent the minimum and the maximum gravity level predicted by the segmentation DNN. Note that, while l_{min} and l_{max} range in $[0, 3]$, the segmentation DNN can also detect the absence of LUS patterns (i.e. the pixel is assigned a score of -1). For this reason, s_{min} and s_{max} range in $[-1, 3]$.

2.3. Evolutionary settings

We use Grammatical Evolution (GE) [23] to evolve programs that resemble DTs (i.e., they are based on an if-then-else structure). GE is an evolutionary algorithm that allows the evolution of grammars, encoded in the Backus-Naur form. It makes use of a genotype, which consists in a list of integers (called *codons*). When the genotype has to be evaluated, it makes use of a translator, which allows to convert the grammar to the corresponding phenotype. The grammar we employ is shown in Table 1.

We consider two GE settings, namely: 1) a single-objective one, in which we optimize either the video-level MSE, the exam-level MSE, or the prognostic-level agreement (see Section 2.3.1 for details on the three metrics); and 2) a multi-objective one, in which we optimize simultaneously all the three objectives stated previously.

The pseudo-code of the algorithm is shown in Algorithm 1. The algorithm consists in an initialization step (Line 1) followed by an evolutionary loop (Lines 4 - 10). The evolutionary loop starts with the evaluation of the population (Line 4), followed by the replacement of the individuals in the population (Line 5). Then, we performs the usual evolutionary steps, i.e., selection (Line 6), crossover (Line 7), and mutation (Line 8).

We should note that the GE algorithm we use here has some differences with respect to the original one described in [23]. First of all, we do not make use of a variable-length genotype but, instead, we fix its length (as shown in Table 2). Fixing the length of the genotype to small values constrain the resulting DTs to be small and, thus, more interpretable than the ones we can obtain by having longer genotypes. Moreover, instead of using the genetic operators described in [23], we use traditional operators for genetic algorithms. The reason underlying this choice is due to the fact that, from preliminary experiments, the original operators seem to achieve worse performance

Algorithm 1: Evolutionary process for the optimization of DTs

```
   Input: $s_p$: the size of the population
   Input: $g$: the number of generations
   Result: $pop$: The final population
1  $pop \leftarrow create\_population(s_p)$;
2  $old\_pop \leftarrow []$;
3  for $i = 0; i < g; i{+}{+}$ do
4      $fitnesses \leftarrow evaluate(pop)$;
5      $pop \leftarrow replacement(pop, old\_pop, fitnesses)$;
6      $parents \leftarrow select(pop, fitnesses)$;
7      $offspring \leftarrow crossover(parents)$;
8      $offspring \leftarrow mutation(offspring)$;
9      $old\_pop \leftarrow pop$;
10     $pop \leftarrow offspring$;
11 end
12 return $pop$;
```

than traditional genetic operators. For this reason, we employ standard operators, described below.

We make use of the replacement operator (Line 5) described in [24], which replaces a parent from the population only if there is an offspring that outperforms it. In case there are two offspring whose performance are better than only one of the two parents, then the best offspring replaces the worst parent.

Moreover, we use of two different parent-selection operators (for Line 6), depending on whether we are working in the single-objective or multi-objective setting. In the single-objective setting, we use the "best-wise" selection operator, i.e., a selection operator that reorders the population by descending fitness such that, when performing crossover, the $(2i)$-th best mates with the $(2i+1)$-th best. Conversely, in the multi-objective setting, the selection operator we use is the NSGA-II [25] operator, which proved to work very well for multi-objective problems[1].

The crossover operator (Line 7), instead, is the one-point crossover, which produces two offspring from two parents by splitting their genotypes in a randomly chosen point and mixing the corresponding sub-strings obtained from the two parents.

While mutating a solution (Line 8), we employ a uniform mutation, which mutates each codon of the genotype according to a given probability p_{codon}. Its new value is sampled randomly from the possible values.

The parameters we use are presented in Table 2. The parameters shown in the table were obtained by manual tuning.

2.3.1. Fitness evaluation

The fitness evaluation phase works as follows. For each training fold, we feed the features of each video to the DT and record its predictions. Then, we compute the following metrics of interest:

1. Video-level MSE (to be minimized):

$$\frac{1}{N_{videos}} \sum_{i=1}^{N_{videos}} (y_i^v - \hat{y}_i^v)^2; \tag{1}$$

[1] Since the most "important" metric is the prognostic-level agreement, an attentive reader may point out that using Pareto optimization may not be the ideal choice in this case. However, we cannot use a lexicographic selection, since this would require specifying a *preference* also between the exam- and the video- level MSE while, in this case, we do not have a clear preference. Moreover, using a weighted sum of the three objectives is also not feasible, since this would require assigning a specific weight to each of the three objectives. For these reasons, we optimize the objectives using Pareto optimization and, then, we select the best solution according to the prognostic-level agreement.

Rule	**Production**
dt	$\langle if \rangle$
if	$if\ \langle condition \rangle\ then\ \langle output \rangle\ else\ \langle output \rangle$
condition	$\langle var \rangle \langle op \rangle \langle const \rangle \mid \langle var \rangle \langle op \rangle \langle var \rangle$
var	$\{input_i\};\ i \in [0, 12[$
op	$< \mid > \mid ==$
output	$0 \mid 1 \mid 2 \mid 3 \mid \langle if \rangle$
const	$[0, 1]$ with step 10^{-2}

Table 1: Grammar used to evolve the DTs. The symbol “|" denotes the possibility to choose between different symbols. When using the grammar to translate a genotype into a phenotype, the rules are expanded in one of the possible choices listed in their production, depending on the value of the genotype.

Parameter	**Value**
Pop size	1000
Generations	1000
Genotype length	50
Crossover probability	0.8
Mutation probability	1
Crossover	One-point
Mutation	Uniform with $p_{codon} = 0.05$
Selection	Best

Table 2: Parameters used for the Grammatical Evolution algorithm.

2. Exam-level MSE (to be minimized):

$$\frac{1}{N_{exams}} \sum_{j=1}^{N_{exams}} (y_j^e - \hat{y}_j^e)^2; \tag{2}$$

3. Prognostic-level agreement (to be maximized):

$$\frac{1}{N_{exams}} \sum_{j=1}^{N_{exams}} \mathbb{I}(y_j^p = \hat{y}_j^p)^2; \tag{3}$$

where: y_i^v is the ground truth for the video i; $y_j^e = \sum_{i=1}^{14} y_{j,i}^v$ is the ground truth for exam j, i.e., the sum of the scores of each video of the exam; $y_j^p = \mathbb{I}(y_j^e > 24)$ is the ground truth for the prognosis j; $\mathbb{I}$ is the indicator function, i.e., it outputs 1 if the argument is true, otherwise 0. The notation $\hat{y}_b^a$ refers to the output of the DT given the output b in setting a, i.e., it is the approximation made by the DT of the variable y_b^a.

The pseudo-code for the fitness evaluation function (in the most general case, i.e., multi-objective) is shown in Algorithm 2. In the pseudo-code, a lowercase bold variable represents a vector, while an uppercase bold variable represents a matrix. Otherwise the variable is assumed to be scalar.

The reason underlying the optimization of different metrics is the following. Our overall goal is to maximize the agreement for all the three metrics, as done in [22]. However, as we discussed earlier optimizing the video- and exam-level agreement requires also a specification of the tolerances to use for the computation of the agreement (e.g., in [22], the authors compute the exam-level agreement with a tolerance of 2, 5, and 10 points). Instead, by optimizing the MSE for these two metrics (Lines 6-7 of Algorithm 2) allows us to evolve DTs that minimize the distance of the predictions from the ground truth, no matter the threshold. Finally, for the prognostic-level agreement, we cannot use the MSE because this variable is not a score but, instead, it is a binary variable. So, in this case, using the MSE does not give any advantage over directly optimizing the agreement (Line 8).

Then, for each metric, we use as fitness the worst value obtained on the 4 folds used for training (Lines 11-13).

While the single-objective fitness corresponds to a scalar value that consists in the value of a single metric, in the multi-objective setting it is composed of a list of three values, i.e., the the video-level MSE, the exam-level MSE, and the prognostic-level agreement.

3. Results

We perform 10 independent runs for the proposed method in each of the four settings: single-objective, video-level MSE; single-objective, exam-level MSE; single-objective, prognostic-level agreement; multi-objective. For each run, we test (on the test fold) only the best evolved DT, i.e., for the single-objective runs it is the individual with the best fitness, while for the multi-objective runs it is the one with the maximum prognostic-level agreement (i.e., the most important metric).

Tables 3, 5, 7 show the descriptive statistics computed on the agreement (%) (with the physicians opinion) computed across all the 5 folds. Bold values represent the best score across all the methods. We compute the statistics on all the 5 folds for the following reason. Since this work is meant to work in a medical scenario, we are not really interested in knowing the training and the test agreements as such. Instead, we are interested in knowing the worst-case scenario and, to compute it, we need to compute the statistic on all the 5 folds. Note that, for each fold, we compute the three agreement scores on that fold and then we use these scores to compute the statistics across folds. In the tables,

Algorithm 2: Fitness evaluation function (multi-objective case)

```
Input: $T$: the DT to evaluate
Input: $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, \mathbf{X}_4$: input features of each fold
Input: $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4$: video-level ground truth
Input: $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \mathbf{e}_4$: exam-level ground truth
Input: $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4$: prognostic-level ground truth
Result: $f$: a list of fitnesses
$num\_folds \leftarrow 4$ // Number of folds used for the training
$e_v \leftarrow []$;
$e_e \leftarrow []$;
$a_p \leftarrow []$;
// Iterate over the folds
for $(i = 0; i < num\_folds; i{+}{+})$ do
    // For each fold, compute the metrics and concatenate
    $e_v \leftarrow concatenate(e_v, [video\_mse(T, \mathbf{X_i}, \mathbf{v_i})])$ // Eq. 1
    $e_e \leftarrow concatenate(e_e, [exam\_mse(T, \mathbf{X_i}, \mathbf{e_i})])$ // Eq. 2
    $a_p \leftarrow concatenate(a_p, [prognostic\_agreement(T, \mathbf{X_i}, \mathbf{p_i})])$ // Eq. 3
end
// Assign the worst-case to each metric
$f \leftarrow []$;
$f[0] \leftarrow max(e_v)$;
$f[1] \leftarrow max(e_e)$;
$f[2] \leftarrow min(a_p)$;
return $f$;
```

"Video", "Exam", "Prognostic" and "3 objectives" refer to, respectively, our method evolved on video, exam, prognostic and three objectives. In the same tables, we also report the state-of-the-art results presented in [22] , listed as "JASA L" and "JASA L+S", which refer to, respectively, the approach using only the labeling DNN, and the one using both the labeling and the segmentation DNNs. Note that we do not use the results shown in [22] but rather we evaluate them on the same folds used for the DTs, to guarantee a uniform evaluation of all the methods.

Finally, it should be considered that even if we use MSE as the metric for the first two objectives, we are still interested in evaluating the agreement with the physicians. For this reason, we do not show the MSE in the tables, but the agreement at video- and exam-level. This allows us also to use the same thresholds as in [22], keeping a consistency on the method used for evaluating such models.

Analyzing the results obtained in the single-objective setting, we observe that, for each metric, the model that achieves the maximum performance (especially for higher thresholds) is the one that has been specifically evolved for that metric. On the other hand, we observe that the model evolved on three objectives in some cases has a smaller minimum/mean agreement than that of the model specifically evolved on each metric. However, it is never smaller than the minimum/mean agreement of the other two models evolved on the other metrics. This suggests that the model evolved by means of multi-objective optimization has a good trade-off between the three objectives.

Surprisingly, we observe that in some cases the performance of the DTs evolved in the multi-objective setting (i.e., the ones evolved on the three objectives simultaneously) exceeds even the performance of the best DTs found in the single-objective setting. This suggests that optimizing for all the metrics simultaneously can keep a "consistency" between the different metrics and allows the DT to learn better strategies for classifying the samples. In fact, we find that the DTs evolved on single objectives do not generalize well to the other objectives. Instead, the DT evolved in the multi-objective setting is able to keep a good trade-off between the objectives.

In Tables 4, 6 and 8 show the results of a statistical comparison performed with a Welch T-test with confidence level $\alpha = 0.05$. The "+", "=", and "-" in the tables indicate respectively statistically better, equal or worse performance of the method on the row w.r.t. the method on the column. We observe that we can reject the null hypotheses (i.e., that the samples come from the same distribution) in just a few cases, namely:

- the DT evolved on the video-level MSE outperforms all the other approaches in the setting with threshold 1;
- the DT evolved with the multi-objective approach outperforms the other approaches in three cases.

In all the other case, the methods result statistically equivalent. However, we should note that this lack of statistical significance is likely due to the small number of folds (i.e., samples for the T-test), even though this has been manually tuned to balance fold size (to reduce overfitting and assess better the generalization capabilities) and number of folds.

Finally, compared with the two methods described in [22] , we observe that while the DTs evolved in the multi-objective setting have a comparable (usually better) minimum agreement than that of JASA L and JASA L+S, they perform substantially better when considering the mean agreement computed on the folds. Also in this case, due to the small number of folds, we can statistically confirm only the increase in performance w.r.t. JASA L+S in the exam-level mean agreement when using a threshold of 10.

Table 3: Descriptive statistics of the video-level agreement on all the folds. "Th" stands for the video-level threshold (i.e., the tolerance) used for the evaluation of the results.

Method	**Th**	**Min**	**Mean**	**Std**	**Med**	**Max**
JASA L	0	44.70	**50.40**	4.28	50.68	**57.38**
	1	82.74	85.87	2.49	85.91	89.76
JASA L+S	0	42.35	50.10	5.48	**51.43**	56.67
	1	82.74	85.87	2.43	85.36	89.29
Video	0	42.75	46.08	2.41	46.07	49.88
	1	**88.63**	**92.40**	**2.14**	**93.41**	**94.52**
Exam	0	42.35	47.82	4.74	46.14	54.29
	1	80.78	84.42	3.00	85.91	87.86
Prognostic	0	41.18	48.21	4.46	50.68	52.62
	1	79.61	83.71	2.79	85.48	86.43
3 objectives	0	**45.88**	49.48	**2.24**	49.77	52.86
	1	85.47	88.31	2.29	87.84	92.14

Table 4: Statistical comparison of the different approaches on the video-level agreement. "+" means "statistically better", "=" means "statistically equivalent", and "-" means "statistically worse" (Welch T-test, $\alpha = 0.05$).

Method	Th	JASA L	JASA L+S	Video	Exam	Prognostic	3 objectives
JASA L	0	=	=	=	=	=	=
	1	=	=	-	=	=	=
JASA L+S	0	=	=	=	=	=	=
	1	=	=	-	=	=	=
Video	0	=	=	=	=	=	-
	1	+	+	=	+	+	+
Exam	0	=	=	=	=	=	=
	1	=	=	-	=	=	=
Prognostic	0	=	=	=	=	=	=
	1	=	=	-	=	=	-
3 objectives	0	=	=	+	=	=	=
	1	=	=	-	=	+	=

4. Analysis of the decision trees

In this section, we show the best evolved DTs in each setting and interpret them to understand the relationships they captured on the predictions made by the DNNs. We consider as "best DT" the

Table 5: Descriptive statistics of the exam-level agreement on all the folds. “Th” stands for the exam-level threshold (i.e., the tolerance) used for the evaluation of the results.

Method	Th	Min	Mean	Std	Med	Max
JASA L	2	21.05	32.30	6.75	32.26	41.94
	5	45.00	56.95	9.43	57.90	70.97
	10	80.00	89.44	6.44	89.47	**100.00**
JASA L+S	2	21.05	30.97	7.58	35.00	38.71
	5	45.16	59.44	13.23	52.63	**80.64**
	10	80.00	84.58	6.27	81.25	96.77
Video	2	21.05	29.65	8.41	25.81	45.16
	5	40.00	57.75	12.14	**63.16**	70.97
	10	80.00	85.85	**4.47**	84.38	93.55
Exam	2	25.00	31.90	7.89	29.03	**46.88**
	5	**61.29**	**64.27**	**3.28**	**63.16**	70.00
	10	73.68	88.01	7.62	**90.32**	95.00
Prognostic	2	**26.32**	30.38	**3.77**	29.03	37.50
	5	45.00	58.50	8.81	59.38	67.74
	10	78.95	87.06	4.96	87.50	93.55
3 objectives	2	21.05	**36.36**	9.68	**38.71**	**46.88**
	5	45.00	63.98	11.35	62.50	77.42
	10	**85.00**	**91.51**	5.34	**90.32**	**100.00**

trees that satisfy the following properties (over the best solutions obtained in the 10 runs). For the setup considering only the video-level MSE, the best tree is the one that obtains the smallest MSE. When we only consider exam-level MSE, the best tree is the tree that achieves the smallest exam-level MSE. When we consider only the prognostic-level agreement, the best tree is the one with the highest prognostic-level agreement. Finally, when we consider all the objectives simultaneously, the best tree is the one that achieves the best prognostic-level agreement. In this case, if there are ties between two solutions, we choose the one that has the best trade-off between video- and exam- level MSE. In all the cases, the conditions of the DT are numbered as when doing a pre-order traversal of the DT.

4.1. Decision tree evolved on the video-level MSE

Figure 5 shows the best DT obtained in this setting. While this DT performs worse than JASA L and JASA L+S when no tolerance is given to the prediction, it outperforms them significantly when a threshold of 1 is allowed. For this reason, we will interpret it considering that each prediction $\hat{y}$ must be considered as a value ranging in $[\hat{y} - 1, \hat{y} + 1]$ (constraining the values in $[0, 3]$).

This DT checks very few things about the predictions made by the two DNNs. In fact, in the first

Table 6: Statistical comparison of the different approaches on the exam-level agreement. "+" means "statistically better", "=" means "statistically equivalent", and "-" means "statistically worse" (Welch T-test, $\alpha = 0.05$).

Method	Th	JASA L	JASA L+S	Video	Exam	Prognostic	3 objectives
JASA L	2	=	=	=	=	=	=
	5	=	=	=	=	=	=
	10	=	=	=	=	=	=
JASA L+S	2	=	=	=	=	=	=
	5	=	=	=	=	=	=
	10	=	=	=	=	=	-
Video	2	=	=	=	=	=	=
	5	=	=	=	=	=	=
	10	=	=	=	=	=	=
Exam	2	=	=	=	=	=	=
	5	=	=	=	=	=	=
	10	=	=	=	=	=	=
Prognostic	2	=	=	=	=	=	=
	5	=	=	=	=	=	=
	10	=	=	=	=	=	=
3 objectives	2	=	=	=	=	=	=
	5	=	=	=	=	=	=
	10	=	+	=	=	=	=

Table 7: Descriptive statistics of the prognostic-level agreement on all the folds. "Th" stands for the (prognostic) threshold used for the evaluation of the results.

Method	**Th**	**Min**	**Mean**	**Std**	**Med**	**Max**
JASA L	24	63.13	78.12	7.96	80.64	85.00
JASA L+S	24	57.90	76.78	12.01	83.87	90.00
Video	24	57.89	79.63	11.68	**85.00**	90.32
Exam	24	61.29	76.25	12.76	77.42	**95.00**
Prognostic	24	**73.68**	81.89	**5.27**	81.25	90.00
3 objectives	24	68.42	**82.11**	7.51	84.38	90.00

split ($l_{max} > s_{max}$) it captures a very simple pattern: when the maximum class of risk predicted by the labeling DNN is greater than the maximum risk class predicted by the segmentation DNN, then it assigns the video a risk varying in $[0, 2]$, i.e., it excludes the class 3. On the other hand, when the root condition is false, it checks whether the fraction of frames classified as maximum risk (by the labeling DNN) is bigger than the fraction of predictions made by the segmentation DNN in

Table 8: Statistical comparison of the different approaches on the prognostic-level agreement. "+" means "statistically better", "=" means "statistically equivalent", and "-" means "statistically worse" (Welch T-test, $\alpha = 0.05$).

Method	Th	JASA L	JASA L+S	Video	Exam	Prognostic	3 objectives
L	24	=	=	=	=	=	=
L+S	24	=	=	=	=	=	=
Video	24	=	=	=	=	=	=
Exam	24	=	=	=	=	=	=
Prognostic	24	=	=	=	=	=	=
3 objectives	24	=	=	=	=	=	=

which the highest score is 2 ($s_2 < l_3$). If so, it assigns the video a score varying in $[2, 3]$, i.e., high risk. Basically, this condition checks whether the video refers to a high-risk patient. In fact, the condition can be interpreted as:

> *If the ratio of samples classified as maximum risk by the labeling DNN is bigger than the ratio of samples classified as risk 2 by the segmentation DNN, then give the priority to the labeling DNN and assign the maximum score to the video.*

To confirm this hypothesis, in Figure 6c we plot the histogram of the number of videos assigned to each class that fall in the case explained above (note that in the other sub-figures of Figure 6 we do the same for all the other conditions in the DT). We observe that the number of videos belonging to class 3 is significantly higher w.r.t. the other classes. Finally, in the third condition ($l_2 < 0.15$) the DT makes an extremely simple check:

> *If the ratio of frame labeled with class 2 is low (i.e., under a threshold of 0.15), then probably the number of frames assigned to class 3 will be even lower, so assign a score ranging in* $[0, 2]$ *to the video. Otherwise, there is a high chance that the severity score is higher than 0, so assign a score ranging in* $[1, 3]$ *to the video.*

From this DT, we infer that the labeling DNN may be "biased" towards high scores. The DT is then evolved to make use of the output of the segmentation DNN in order to reduce this bias.

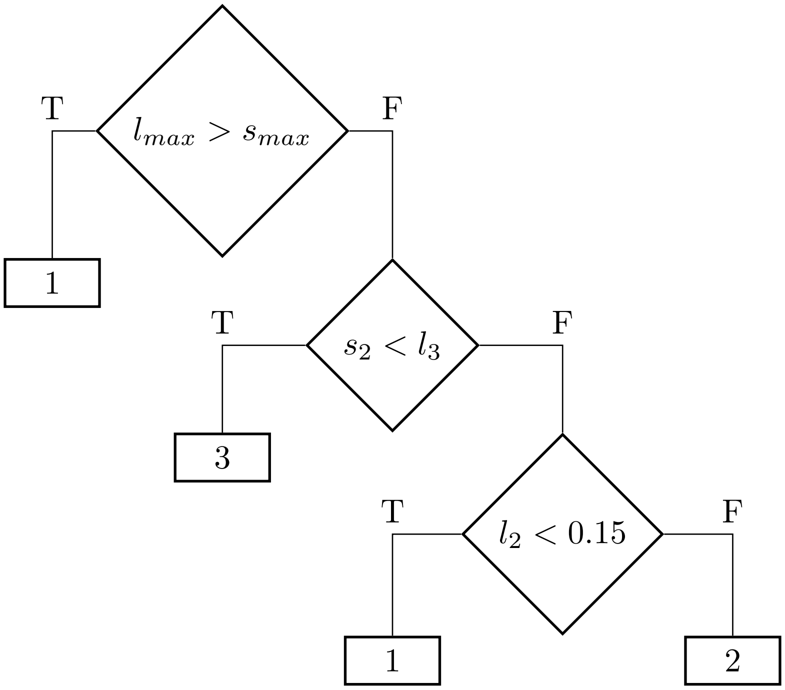


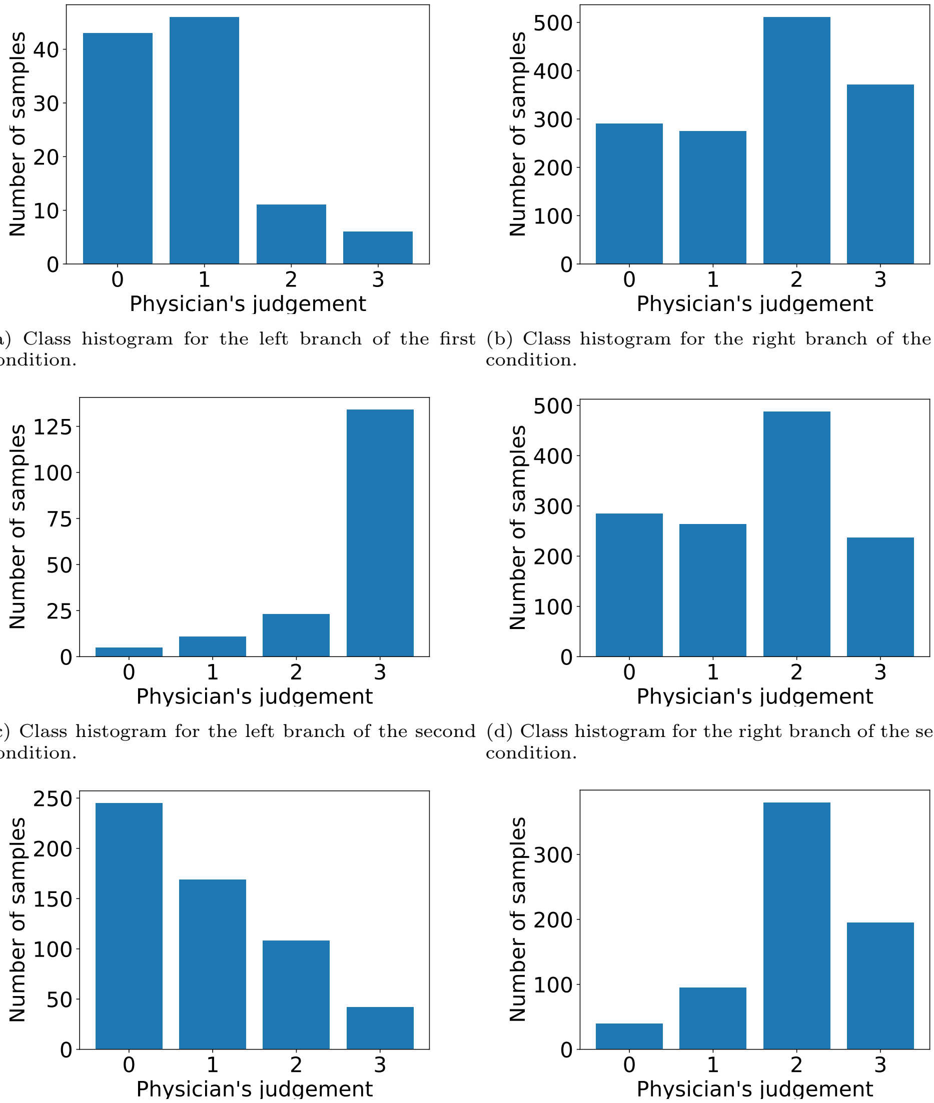
Figure 5: Best DT evolved on the video-level MSE.

4.2. Decision tree evolved on the exam-level MSE

This DT (shown in Figure 7) achieves a better worst-case agreement with low thresholds (2 and 5). However, in this case (and the following ones), we cannot use the threshold as a tolerance value to be used on the output value of the DT. This is due to the fact that, in this case, the DT outputs the gravity for each video, but the tolerance is expressed at the exam level.

The first condition of this DT ($l_0 < 0.72$) checks that the severity of the patient is high, by ensuring that the fraction of frames labeled as minimum risk is lower than an evolved threshold. If so, it then assesses the severity of the conditions by using the segmentation DNN, checking if the fraction of frames that are classified as maximum risk is more than the half ($s_3 > 0.51$). If so, it assigns the maximum risk to the video.

If the first condition is false, then the DT performs additional checks. In fact, the right part of the DT is basically a decision list, i.e., an extremely unbalanced DT, which isolates one particular case at each split. While the first condition on the right ($s_2 > l_{max}$) may make no sense at a first sight, it is a simple trick that the DT uses to perform an *and* between two conditions. In fact, we



(a) Class histogram for the left branch of the first condition.

(b) Class histogram for the right branch of the first condition.

(c) Class histogram for the left branch of the second condition.

(d) Class histogram for the right branch of the second condition.

(e) Class histogram for the left branch of the third condition.

(f) Class histogram for the right branch of the third condition.

Figure 6: Class histograms for each of the branches of the DT shown in Figure 5. Note that the nodes are counted as in a pre-order traversal of the DT.

know that $s_2 \in [0, 1]$ and $l_{max} \in \{0, 1, 2, 3\}$. This means that the condition $s_2 > l_{max}$ evaluates to true only in case $s_2 > 0$ and $l_{max} = 0$.

The second case isolated by the decision list checks for a particular case ($s_1 = l_3$). By analyzing the training set, we observe that this case only happens when $s_1 = l_3 = 0$. Moreover, in these cases s_1 and l_3 are the only variables that are always equal to zero. While this may seem a remote possibility, we found that this condition (conjoined with the conditions that are evaluated before it) evaluates to true for about a quarter of the samples in the training set. Hence, this condition exploits a bias of the two DNNs to detect cases in which the severity is likely to be low (45% of the cases with score 0, 31% of the cases with score 1, 17% with score 2, 7% with score 3).

The third condition on the right branch ($s_{min} = 0$) checks whether the patient has at least one frame with minimum risk (i.e., 0, opposed to a case in which no damaged tissue is detected, i.e., -1) by checking the outputs of the segmentation DNN. If so, it assigns the class 2 to the video.

Finally, the last condition ($s_0 > 0.37$) checks the number of frames with minimum risk (detected by the segmentation model): if they consist of more than the 37% of the frames in the video, the risk assigned to the video is very high (3), otherwise it is assigned a lower score (1).

It is important to note that the video-level predictions performed by this model aim to reduce the worst-case exam-level MSE w.r.t. the physicians' judgment.

Also in this case, for the sake of completeness we report in Figure 8 the distribution of classes for each condition in the DT.

4.3. Decision tree evolved on the prognostic-level agreement

This DT (shown in Figure 9) achieves the best worst-case prognostic-level agreement among all the best evolved DTs (in this case, evolved with the single threshold value, 24).

In the root condition ($l_3 < s_1$), this DT checks whether the confidence given to class 3 from the labeling DNN is smaller than the confidence given to class 1 by the segmentation DNN. This, intuitively, tries to filter out the cases where the probability of having the maximum risk is high. In fact, as shown in Figure 10a, the ratio of samples belonging to class 3 is not so high in this case (12.5%), see the other sub-figures in Figure 10 for the distributions of classes corresponding to the other conditions in the DT.

The second condition (i.e., the left branch of the root, $l_0 > 0.75$) naturally follows the first one: given that, as shown in Figure 10a, the distribution of the classes is skewed towards class 2, is there a way to filter out the samples belonging to class 2? While this condition does not filter perfectly the samples belonging to class 2, it is able to filter 67.9% of them (as shown in Figures 10c and 10d).

The third condition ($l_{max} = l_{min}$) seeks for cases in which the maximum class and the minimum class predicted by the labeling DNN are equal. Of course, this condition is way more likely to happen

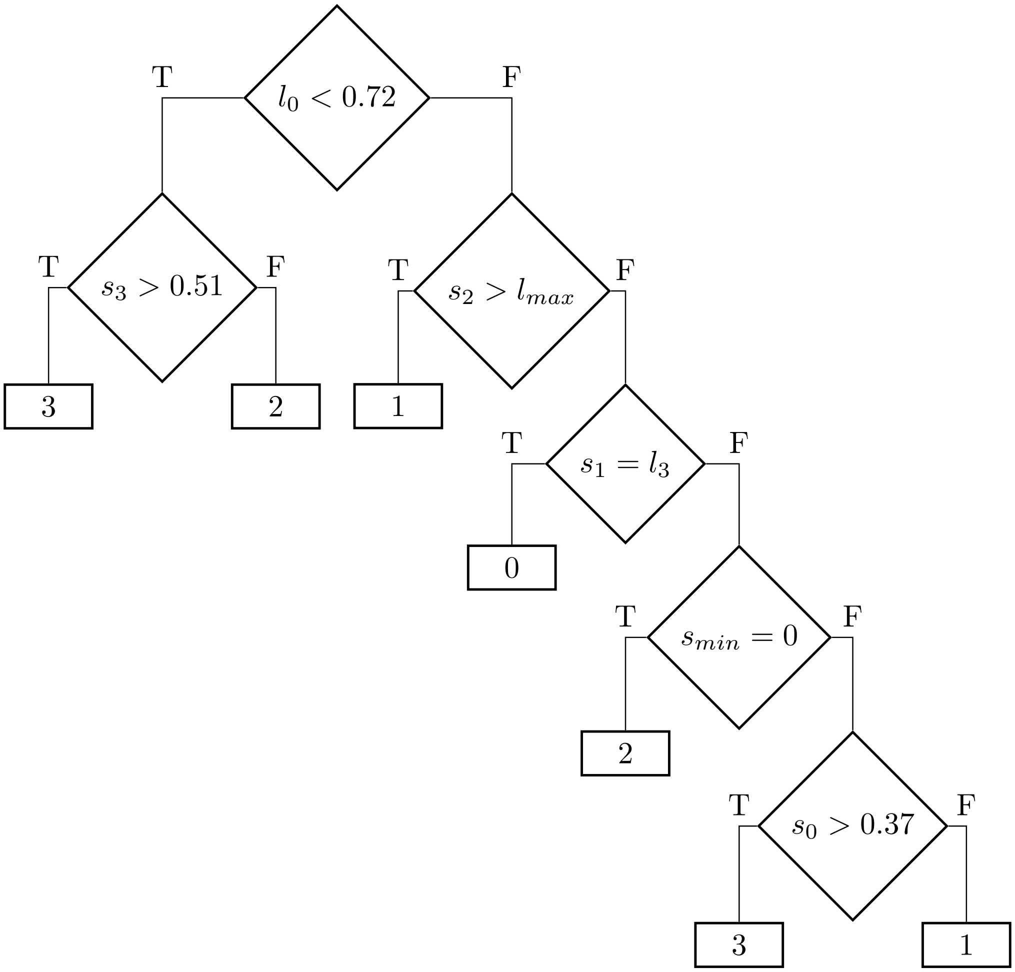
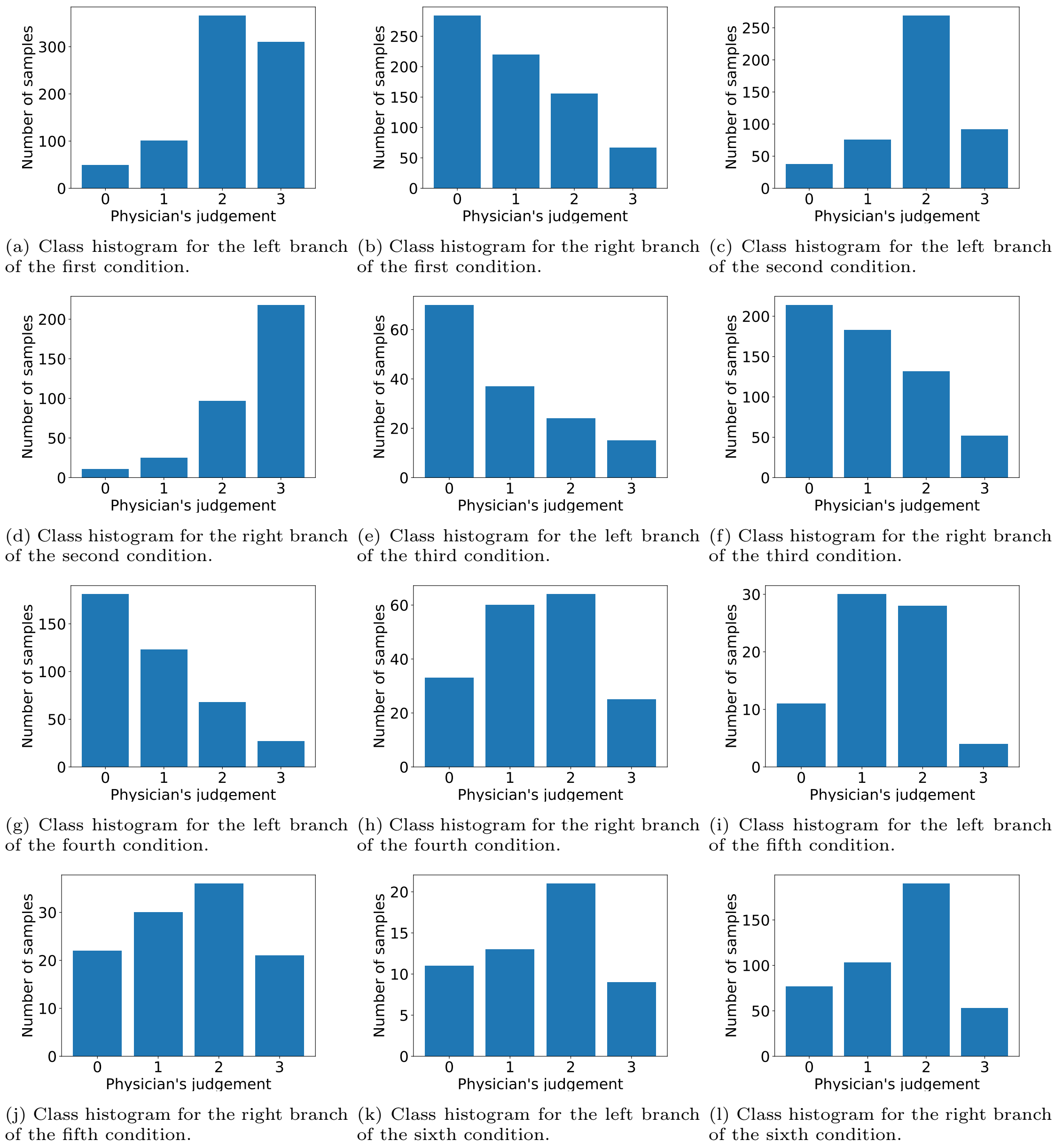


Figure 7: Best DT evolved on the exam-level MSE.

in low-risk frames, as confirmed by Figure 10e. However, as we can see from Figure 10f, not all the samples with low risk are filtered out by this condition.

For this reason, the purpose of the fourth condition ($s_3 < s_0$) is to separate the low-risk cases from the higher-risk ones. In fact, what it does is simply checking the predictions made by the segmentation DNN: if the ratio of samples assigned to class 0 is higher than the ratio of samples assigned to class 3, then it predicts 0, otherwise 2.

Note that this DT has been optimized to maximize the prognostic-level agreement. This explains why, in some cases, the outputs are not coherent with what a human expects when trying to predict the label for each *video*. In fact, we hypothesize that these counter-intuitive tests aim to soften the



(a) Class histogram for the left branch of the first condition.

(b) Class histogram for the right branch of the first condition.

(c) Class histogram for the left branch of the second condition.

(d) Class histogram for the right branch of the second condition.

(e) Class histogram for the left branch of the third condition.

(f) Class histogram for the right branch of the third condition.

(g) Class histogram for the left branch of the fourth condition.

(h) Class histogram for the right branch of the fourth condition.

(i) Class histogram for the left branch of the fifth condition.

(j) Class histogram for the right branch of the fifth condition.

(k) Class histogram for the left branch of the sixth condition.

(l) Class histogram for the right branch of the sixth condition.

Figure 8: Class histograms for each of the branches of the DT shown in Figure 7. Note that the nodes are counted as in a pre-order traversal of the DT.

contributions of each video to the prognostic score. This can be seen in the fact that predictions for the class 3 never appear in this DT, and also that the predictions for class 1 are not frequent (even when they would minimize the video-level MSE, which is not taken into account when optimizing this DT).

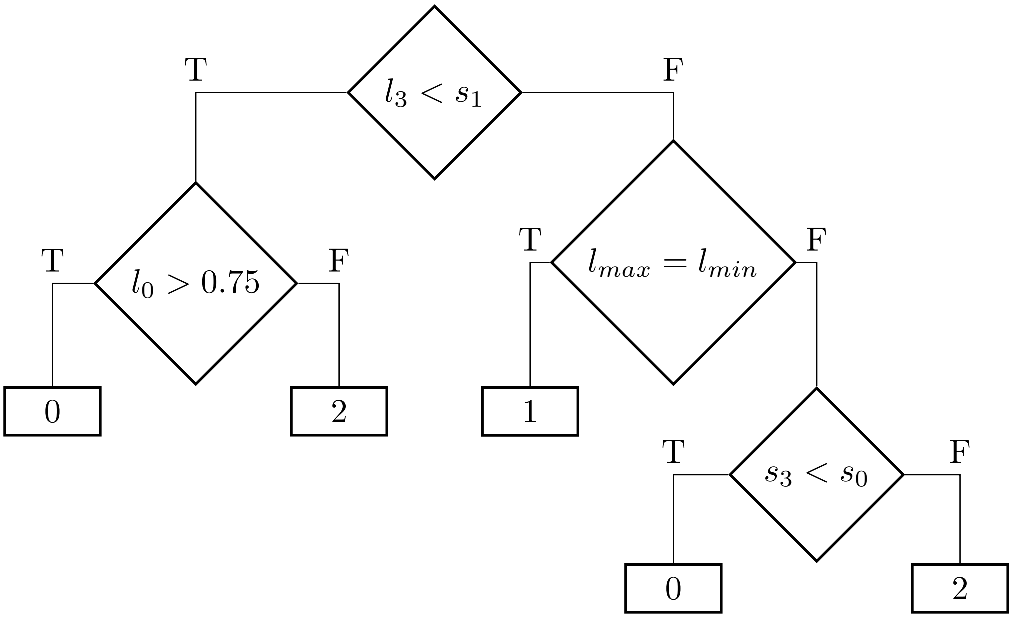


Figure 9: Best DT evolved on the prognostic-level agreement.

4.4. Decision tree evolved on three objectives

This DT (shown in Figure 11) has comparable, but often better, performance with respect to all the other DTs evolved in the other settings. One interesting feature of this DT is that it uses the labeling DNN to make coarse-grained decisions, that are then refined by using the segmentation DNN.

In the first condition ($l_3 > l_0$), this DT simply checks the outputs coming from the labeling DNN to address the gravity of the conditions. Surprisingly, only checking if the ratio of samples assigned to risk 3 is higher than the ratio of samples assigned to risk 0 is enough to discriminate very well the high-risk cases, as shown in Figure 12a (see the other sub-figures in Figure 12 for the distributions of classes corresponding to the other conditions in the DT).

In the second condition ($l_2 > 0.07$), the DT checks the ratio of labels assigned to class 2 by the labeling DNN. If they are more than 7%, then it makes a simple refinement using the segmentation DNN: if the segmentation DNN classifies all the samples as class 3 ($s_3 = 1$), then it assigns the video the maximum score, otherwise it assigns the video a score of 2.

If the second condition evaluates to false, then the DT checks whether $l_{max} < 0.97$. Since l_{max} is an integer, this corresponds to checking whether $l_{max} = 0$. If so, again, the DT makes use of the segmentation DNN to refine the decision: if the ratio of samples assigned to class 3 by the segmentation DNN is more than the 60% ($s_3 > 0.60$), then it assigns the video a score of 2. Otherwise, the gravity of the condition is not high enough, so it assigns a score 0 to the video. This condition handles a bias of the labeling DNN, that happens when this DNN classifies all the frames with a severity of 0, but, instead, their actual score is very different from 0.

Finally, if $l_{max} > 0.97$, it uses a similar check ($s_3 > 0.59$) to assign the samples either to class 0 or 1. Surprisingly, when s_3 is greater than 59%, the DT assigns the sample to class 0 while, as we can see from Figure 12k, assigning it a value equal to 1 would reduce the video-level MSE. However, this reasoning applies to the video-level predictions, but it may affect negatively the other two metrics. On the other hand, when $s_3 \leq 0.59$, we observe that the probability for class 3 is quite low, so the DT classifies the sample as belonging to class 1, probably to minimize the video-level MSE.

5. Conclusions

The use of LUS techniques to monitor the state of the lungs in COVID-19 patients is spreading, due to its numerous advantages compared to other techniques. Moreover, with COVID-19, the need for an *automatic* diagnosis emerged. For this purpose, several approaches have been proposed to perform automatic COVID-19 patients' evaluation from LUS images [17–20]. A previous approach [22]proposed a combination of two DNNs to increase the overall performance. This approach used an empiric threshold-based approach that, while performing well, did not give any insight on the biases of the DNNs used as input.

In recent years, a new need has emerged at the intersection between AI and healthcare: the need for interpretability [26]. In fact, especially in this domain, the users (in this case, the physicians) usually do not trust decisions suggested by a black box model (such as one base on DDNs). Instead, they want to be able to understand *each* decision made by the model, to ensure its correctness. Interpretable AI allows to have data-driven models that are inherently understandable and "simulatable" by humans, thus ensuring that a physician can actually understand the decisions made by the model.

In this work, we use two previously proposed DNNs as *feature extractors*, and then we use a DT for combining the two predictions. We use both single- and multi-objective evolutionary optimization to evolve the DT that takes in input the predictions made by the two DNNs aggregated at the video-level (i.e., a collection of frames). When evaluating our approach on three different levels of agreement with the physicians' judgment, we find that the multi-objective optimization approach leads to DTs that, in general, perform in most cases comparably or better than the DTs evolved on single objectives. Moreover, our approach appears to perform better (in terms of descriptive statistics) than the approach presented in [22], even though, due to the small number of samples used for the comparison, we were able to quantify the statistical significance of the results only in a small number of cases.

This aspect, i.e., the fact that the limited number of samples, in some cases, does not allow a statistically significant comparison w.r.t. the baseline algorithm from [22], represents one of the main limitations of the current work. Another limitation of this study is that the DTs use orthogonal conditions (i.e., they compare a variable with a constant), or conditions that compare one variable to another variable. Since there are more expressive types of conditions (e.g., oblique conditions), better results may be achievable through the use of different types of conditions. Moreover, considering more complex conditions to describe the relationship between more than two variables may lead to better insights about the biases of the DNNs.

In the light of these limitations, future research directions will be aimed at collecting more data, in order to increase the size of the dataset, and evolving DTs by using different types of conditions, including oblique ones.

Finally, we highlight that we make our data publicly available for further development and reproducibility[2]. Moreover, in a separate repository[3] we release the scripts used to produce the results shown in this paper.

References

[1] G. Soldati, A. Smargiassi, R. Inchingolo, D. Buonsenso, T. Perrone, D. F. Briganti, S. Perlini, E. Torri, A. Mariani, E. E. Mossolani, F. Tursi, F. Mento, L. Demi, Is There a Role for Lung Ultrasound During the COVID-19 Pandemic?, Journal of Ultrasound in Medicine 39 (7) (2020)

[2] https://drive.google.com/drive/folders/1Or4dF2fAM23H5fd_yxtq1vyAS8b7pL0s

[3] https://gitlab.com/leocus/neurosymbolic-covid19-scoring

1459–1462. doi:10.1002/jum.15284.
URL https://doi.org/10.1002/jum.15284

[2] G. Soldati, A. Smargiassi, R. Inchingolo, D. Buonsenso, T. Perrone, D. F. Briganti, S. Perlini, E. Torri, A. Mariani, E. E. Mossolani, F. Tursi, F. Mento, L. Demi, Proposal for International Standardization of the Use of Lung Ultrasound for Patients With COVID-19, Journal of Ultrasound in Medicine 39 (7) (2020) 1413–1419. doi:10.1002/jum.15285.
URL https://doi.org/10.1002/jum.15285

[3] E. Poggiali, A. Dacrema, D. Bastoni, V. Tinelli, E. Demichele, P. Mateo Ramos, T. Marcianò, M. Silva, A. Vercelli, A. Magnacavallo, Can Lung US Help Critical Care Clinicians in the Early Diagnosis of Novel Coronavirus (COVID-19) Pneumonia?, Radiology 295 (3) (2020) E6–E6. doi:10.1148/radiol.2020200847.
URL https://doi.org/10.1148/radiol.2020200847

[4] P. Lomoro, F. Verde, F. Zerboni, I. Simonetti, C. Borghi, C. Fachinetti, A. Natalizi, A. Martegani, COVID-19 pneumonia manifestations at the admission on chest ultrasound, radiographs, and CT: single-center study and comprehensive radiologic literature review, European Journal of Radiology Open 7. doi:10.1016/j.ejro.2020.100231.
URL https://doi.org/10.1016/j.ejro.2020.100231

[5] A. Nouvenne, A. Ticinesi, A. Parise, B. Prati, M. Esposito, V. Cocchi, E. Crisafulli, A. Volpi, S. Rossi, E. G. Bignami, M. Baciarello, E. Brianti, M. Fabi, T. Meschi, Point-of-Care Chest Ultrasonography as a Diagnostic Resource for COVID-19 Outbreak in Nursing Homes, Journal of the American Medical Directors Association 21 (7) (2020) 919–923. doi:10.1016/j.jamda.2020.05.050.
URL https://doi.org/10.1016/j.jamda.2020.05.050

[6] K. Yasukawa, T. Minami, Point-of-Care Lung Ultrasound Findings in Patients with COVID-19 Pneumonia, The American Journal of Tropical Medicine and Hygiene 102 (6) (2020) 1198–1202. doi:10.4269/ajtmh.20-0280.
URL https://www.ajtmh.org/view/journals/tpmd/102/6/article-p1198.xml

[7] C. Xing, Q. Li, H. Du, W. Kang, J. Lian, L. Yuan, Lung ultrasound findings in patients with

COVID-19 pneumonia, Critical Care 24 (1) (2020) 174. doi:10.1186/s13054-020-02876-9.
URL https://doi.org/10.1186/s13054-020-02876-9

[8] Q.-Y. Peng, X.-T. Wang, L.-N. Zhang, C. C. C. U. S. G. (CCUSG), Findings of lung ultrasonography of novel corona virus pneumonia during the 2019–2020 epidemic, Intensive Care Medicine 46 (5) (2020) 849–850. doi:10.1007/s00134-020-05996-6.
URL https://doi.org/10.1007/s00134-020-05996-6

[9] G. Duclos, A. Lopez, M. Leone, L. Zieleskiewicz, "No dose" lung ultrasound correlation with "low dose" CT scan for early diagnosis of SARS-CoV-2 pneumonia, Intensive Care Medicine 46 (6) (2020) 1103–1104. doi:10.1007/s00134-020-06058-7.
URL https://doi.org/10.1007/s00134-020-06058-7

[10] L. Demi, Lung ultrasound: The future ahead and the lessons learned from COVID-19, The Journal of the Acoustical Society of America 148 (4) (2020) 2146–2150. doi:10.1121/10.0002183.
URL https://doi.org/10.1121/10.0002183

[11] M. Allinovi, A. Parise, M. Giacalone, A. Amerio, M. Delsante, A. Odone, A. Franci, F. Gigliotti, S. Amadasi, D. Delmonte, N. Parri, A. Mangia, Lung Ultrasound May Support Diagnosis and Monitoring of COVID-19 Pneumonia, Ultrasound in Medicine and Biology 46 (11) (2020) 2908–2917. doi:10.1016/j.ultrasmedbio.2020.07.018.
URL https://doi.org/10.1016/j.ultrasmedbio.2020.07.018

[12] F. Mento, G. Soldati, R. Prediletto, M. Demi, L. Demi, Quantitative Lung Ultrasound Spectroscopy Applied to the Diagnosis of Pulmonary Fibrosis: The First Clinical Study, IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control 67 (11) (2020) 2265–2273. doi:10.1109/TUFFC.2020.3012289.

[13] F. Mento, L. Demi, On the Influence of Imaging Parameters on Lung Ultrasound B-line Artifacts, in vitro study, Journal of the Acoustical Society of America 148 (2) (2020) 975–983. doi:10.1121/10.0001797.
URL https://asa.scitation.org/doi/abs/10.1121/10.0001797

[14] G. Soldati, M. Demi, The use of lung ultrasound images for the differential diagnosis of

pulmonary and cardiac interstitial pathology, Journal of Ultrasound 20. doi:10.1007/s40477-017-0244-7.

[15] K. Mohanty, J. Blackwell, T. Egan, M. Muller, Characterization of the Lung Parenchyma Using Ultrasound Multiple Scattering, Ultrasound in Medicine and Biology 43 (5) (2017) 993–1003. doi:10.1016/j.ultrasmedbio.2017.01.011.
URL https://doi.org/10.1016/j.ultrasmedbio.2017.01.011

[16] G. Zhang, J. Zhang, B. Wang, X. Zhu, Q. Wang, S. Qiu, Analysis of clinical characteristics and laboratory findings of 95 cases of 2019 novel coronavirus pneumonia in Wuhan, China: a retrospective analysis, Respiratory research 21 (1) (2020) 74. doi:10.1186/s12931-020-01338-8.
URL https://pubmed.ncbi.nlm.nih.gov/32216803https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7099829/

[17] S. Roy, W. Menapace, S. Oei, B. Luijten, E. Fini, C. Saltori, I. Huijben, N. Chennakeshava, F. Mento, A. Sentelli, E. Peschiera, R. Trevisan, G. Maschietto, E. Torri, R. Inchingolo, A. Smargiassi, G. Soldati, P. Rota, A. Passerini, R. J. G. V. Sloun, E. Ricci, L. Demi, Deep learning for classification and localization of COVID-19 markers in point-of-care lung ultrasound, IEEE Transactions on Medical Imaging 39 (8) (2020) 2676–2687. doi:10.1109/TMI.2020.2994459.

[18] L. Carrer, E. Donini, D. Marinelli, M. Zanetti, F. Mento, E. Torri, A. Smargiassi, R. Inchingolo, G. Soldati, L. Demi, F. Bovolo, L. Bruzzone, Automatic Pleural Line Extraction and COVID-19 Scoring from Lung Ultrasound Data, IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control 67 (11) (2020) 2207–2217. doi:10.1109/TUFFC.2020.3005512.

[19] W. Xue, C. Cao, J. Liu, Y. Duan, H. Cao, J. Wang, X. Tao, Z. Chen, M. Wu, J. Zhang, H. Sun, Y. Jin, X. Yang, R. Huang, F. Xiang, Y. Song, M. You, W. Zhang, L. Jiang, Z. Zhang, S. Kong, Y. Tian, L. Zhang, D. Ni, M. Xie, Modality alignment contrastive learning for severity assessment of COVID-19 from lung ultrasound and clinical information, Medical Image Analysis 69. doi:10.1016/j.media.2021.101975.

[20] O. Frank, N. Schipper, M. Vaturi, G. Soldati, A. Smargiassi, R. Inchingolo, E. Torri, T. Perrone, F. Mento, L. Demi, M. Galun, Y. C. Eldar, S. Bagon, Integrating Domain Knowledge into Deep

Networks for Lung Ultrasound with Applications to COVID-19, IEEE Transactions on Medical Imaging (2021) 1doi:10.1109/TMI.2021.3117246.

[21] T. Perrone, G. Soldati, L. Padovini, A. Fiengo, G. Lettieri, U. Sabatini, G. Gori, F. Lepore, M. Garolfi, I. Palumbo, R. Inchingolo, A. Smargiassi, L. Demi, E. E. Mossolani, F. Tursi, C. Klersy, A. Di Sabatino, A New Lung Ultrasound Protocol Able to Predict Worsening in Patients Affected by Severe Acute Respiratory Syndrome Coronavirus 2 Pneumonia, Journal of Ultrasound in Medicinedoi:https://doi.org/10.1002/jum.15548.
URL https://doi.org/10.1002/jum.15548

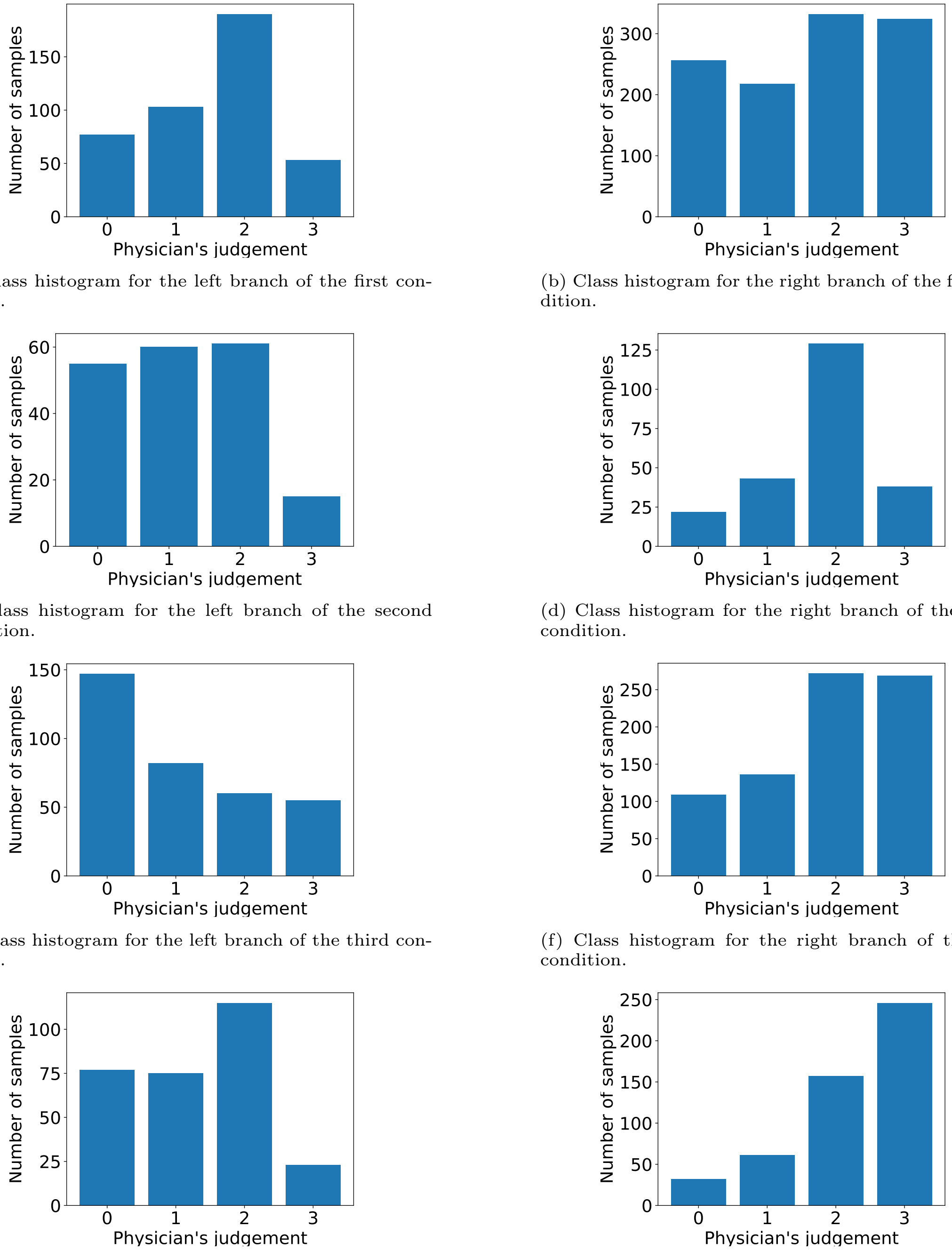
[22] F. Mento, T. Perrone, A. Fiengo, A. Smargiassi, R. Inchingolo, G. Soldati, L. Demi, Deep learning applied to lung ultrasound videos for scoring COVID-19 patients: A multicenter study, The Journal of the Acoustical Society of America 149 (5) (2021) 3626–3634.

[23] C. Ryan, J. Collins, M. O. Neill, Grammatical evolution: Evolving programs for an arbitrary language, in: European Conference on Genetic Programming, Springer, Berlin, Heidelberg, 1998, pp. 83–96. doi:10.1007/BFb0055930.
URL http://link.springer.com/10.1007/BFb0055930

[24] L. L. Custode, G. Iacca, Evolutionary learning of interpretable decision trees (2021). arXiv:2012.07723.

[25] K. Deb, S. Agrawal, A. Pratap, T. Meyarivan, A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II, in: International conference on parallel problem solving from nature, Springer, 2000, pp. 849–858.

[26] A. Vellido, The importance of interpretability and visualization in machine learning for applications in medicine and health care, Neural computing and applications 32 (24) (2020) 18069–18083.



(a) Class histogram for the left branch of the first condition.

(b) Class histogram for the right branch of the first condition.

(c) Class histogram for the left branch of the second condition.

(d) Class histogram for the right branch of the second condition.

(e) Class histogram for the left branch of the third condition.

(f) Class histogram for the right branch of the third condition.

(g) Class histogram for the left branch of the fourth condition.

(h) Class histogram for the right branch of the fourth condition.

Figure 10: Class histograms for each of the branches of the DT shown in Figure 9. Note that the nodes are counted as in a pre-order traversal of the DT.

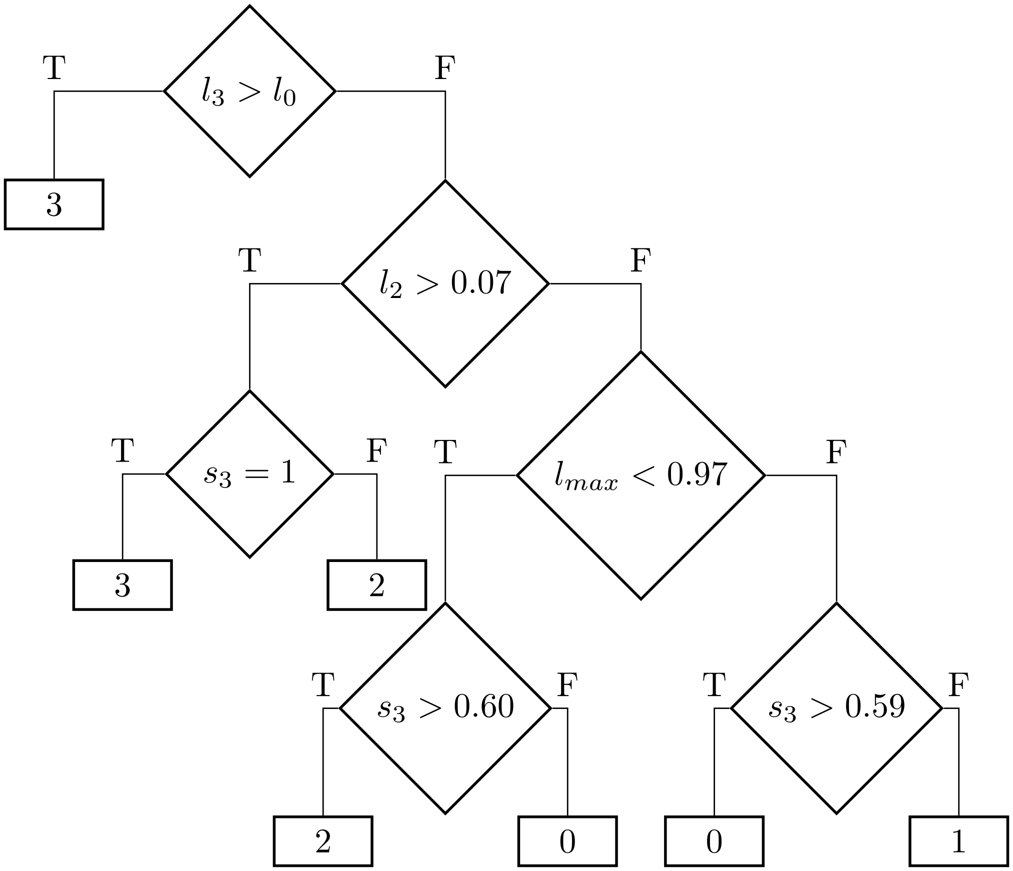
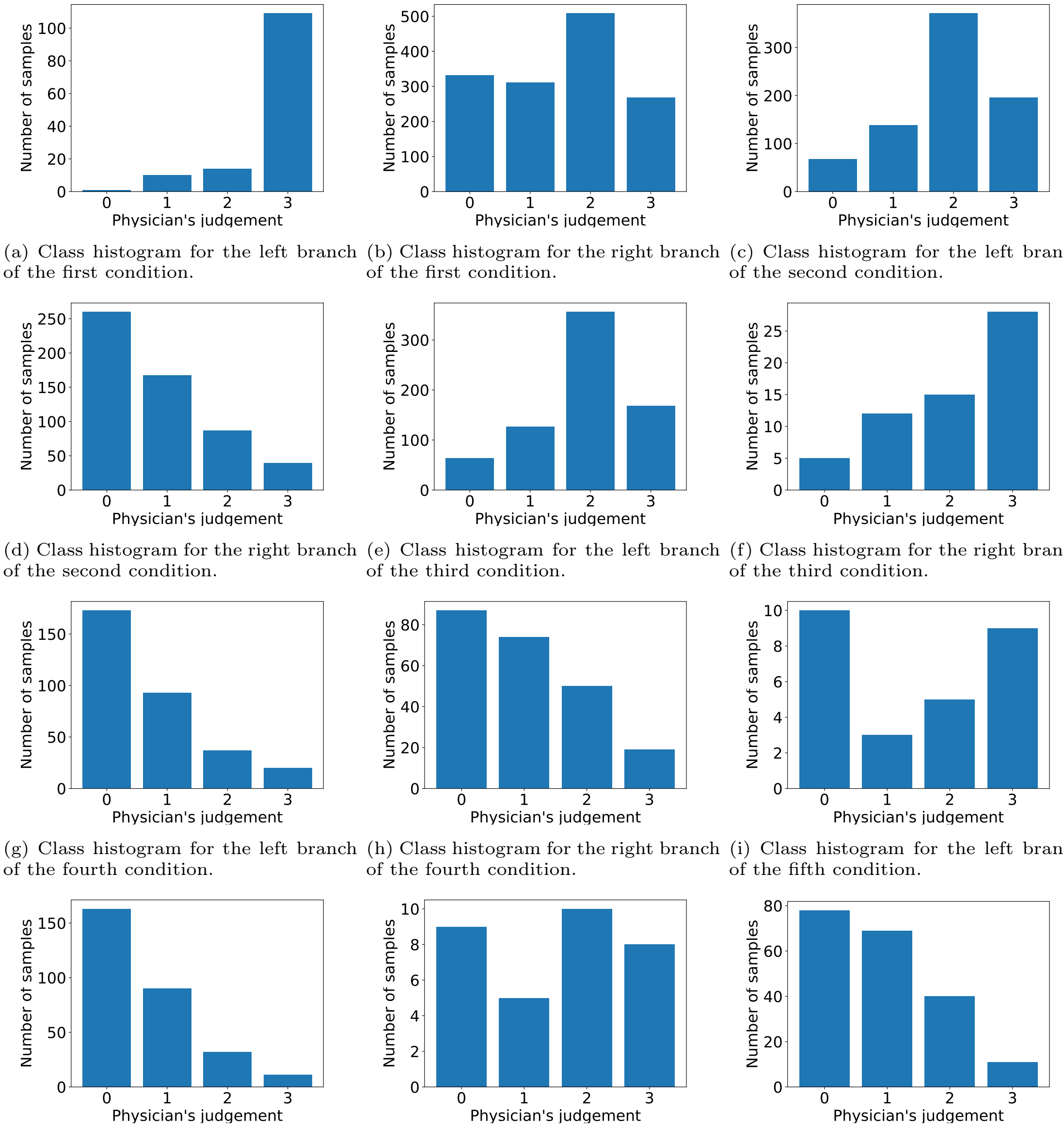


Figure 11: Best DT evolved on the three objectives.



(a) Class histogram for the left branch of the first condition.

(b) Class histogram for the right branch of the first condition.

(c) Class histogram for the left branch of the second condition.

(d) Class histogram for the right branch of the second condition.

(e) Class histogram for the left branch of the third condition.

(f) Class histogram for the right branch of the third condition.

(g) Class histogram for the left branch of the fourth condition.

(h) Class histogram for the right branch of the fourth condition.

(i) Class histogram for the left branch of the fifth condition.

(j) Class histogram for the right branch of the fifth condition.

(k) Class histogram for the left branch of the sixth condition.

(l) Class histogram for the right branch of the sixth condition.

Figure 12: Class histograms for each of the branches of the DT shown in Figure 11. Note that the nodes are counted as in a pre-order traversal of the DT.