# CARING for xAI

Flavio Bertini[1,†], Alessandro Dal Palù[1,*,†], Francesco Fabiano[1,†] and Eleonora Iotti[1,†]

[1]*Department of Mathematical, Physical and Computer Sciences, University of Parma, Parco Area delle Scienze 53/A, 43124, Parma, Italy*

**Abstract**

Over the last few years, *Artificial Intelligence* (AI) has pervaded our lives. As a result, automated tools that "reason" on different scenarios have become more and more common. As this trend continues to grow, it has become necessary to ensure that newly developed tools and technologies can be safely adopted, as demonstrated by the numerous EU regulations. This is especially true when the concept of AI is intertwined with the field of medicine, where every decision may be critical. That is why, in this work, we decided to tackle the problem of *automated interpretation* of *Computed Tomography* (CT) *scans* using an *explainable* approach. In fact, while several methods based on Machine Learning (ML) are currently available, these are still outperformed by medical doctors and provide answers that cannot be traced back to a logical deduction. This paper presents CARING, a new methodology based on Answer Set Programming (ASP), which returns reliable, easy-to-program and explainable interpretations of CT scans. In particular, CARING makes use of transparent technologies in order to handle medical knowledge provided either by experts or by verified ontologies. This proof of concept shows that Logic Programming is a mature technology that can match the newest challenges in the xAI field.

**Keywords**

Explainable AI, Answer Set Programming, CT Scan, Image processing, Tissue Segmentation

## 1. Motivation

This paper investigates the design of CARING (**C**T-scan **A**utomated **R**easoning **IN**terpretation **G**uidance)[1]. The purpose of this tool is to implement an AI pipeline that reproduces the mental process of analysis and interpretation of a CT scan performed by radiologists. The input of the tool is the set of raw data from the acquisition system and the output is the labelling of organs and structures contained in the body scan. The goal of the tool is to provide the basis for further automated analyses and to support Medical Doctor decisions.

The problem contains various challenges that combine artificial vision and reasoning tasks. Even if the problem has been largely studied in the literature, the attempts proposed so far (see Section 2.1) can be classified as black box approaches: the user has little control about accuracy and accountability of system's results; introducing/updating medical information to the system

---

*CILC 2022: 37th Italian Conference on Computational Logic, June 29 – July 1, 2022, Bologna, Italy*

[*] Corresponding author.

[†] These authors contributed equally.

flavio.bertini@unipr.it (F. Bertini); alessandro.dalpalu@unipr.it (A. D. Palù); francesco.fabiano@unipr.it (F. Fabiano); eleonora.iotti@unipr.it (E. Iotti)

0000-0001-6925-5712 (F. Bertini); 0000-0003-0353-158X (A. D. Palù); 0000-0002-1161-0336 (F. Fabiano); 0000-0001-7670-2226 (E. Iotti)

CEUR Workshop Proceedings (CEUR-WS.org)

[1] ahead-lab.unipr.it/caring

require significant rewriting/training. According to recent EU regulations about *explainable AI* (see Section 2.2), Medical Doctors can include an AI based system in their workflow under certain requirements. The most relevant ones are that the system must provide a support or proof for its outputs. We believe that this perspective poses interesting challenges to the Logic Programming (LP) community, since this paradigm naturally supports such requirements.

Radiologists, when interpreting the content of a CT scan, typically start from spatially local details, find relations among regions and eventually recognize all anatomic details. Local details are combined into geometric regions and (higher-level) medical knowledge allows to map relations among local details, in order to support the proposed classification. The process has to cope with a remarkable body's variability: *e.g.*, organs can appear in different sizes, locations and shapes. Structural relationships are by far more preserved, even if several anatomical structural variants are documented (see, *e.g.*, [1] for blood vessel ones).

In this context, details in the scans, or *features*, are represented by visible surfaces that separate regions with different properties (hypothetical boundary between organs/tissues) and uniform regions that may belong to the same object. These features can be easily noticed in the scan, even by an untrained viewer, and they are one of the first points of attention in the analysis performed by radiologists. Each feature does not necessarily correspond to a specific tissue or organ, but this first step of abstraction has the effect of reducing the computational burden of the next phase. Common edge detection and segmentation algorithms (classical off-the-shelf tools, Machine Learning approaches, Neural Networks) often go beyond simple geometric analyses and encode some higher level strategy to resolve some ambiguities to deliver a more accurate result. Unfortunately, such approaches hard code some declarative properties about the domain that can not be easily controlled as high level properties (*e.g.*, there are parameters to be tuned; a trained network provides no understanding of the rules that drive the process). We find here two main issues: (i) the algorithms' output provides no high level explanation of the assumptions made, (ii) it is difficult and/or impossible to modify the behaviour of such algorithms depending on some high level requirements.

Such scenario suggests a processing pipeline where the identification of pure geometric basic elements is decoupled from reasoning about their possible anatomic role. In order to create an explainable reasoning, geometric feature extraction must operate with the lack of any medical information.

The goal is to delay the interpretation process to the reasoning phase. The advantage is that the reasoning is completely in charge of exploiting medical knowledge and it is able to give proof of every hypothesis and deduction process.

The paper presents a background Section 2 with a brief overview on CT scans, explainable AI, medical ontology and LP methods employed. Section 3 introduces a CSP based geometric feature extraction and Section 2.3 describes the ASP based reasoner on geometric features. In Section 5 we show some preliminary results and in Section 6 we conclude.

Let us conclude the introduction with a summary of the advantages of having an LP core inside the pipeline:

- medical knowledge is modelled in a LP framework. It can be handled as a collaborative knowledge base (with possibility to merge inconsistent facts; retrieve legal liability of authors of input knowledge; incremental and distributed management) and/or as ontology;

- medical knowledge can be translated into and from natural language for easier interaction with Medical Doctors;
- the reasoner offers a transparent approach and the output is explainable: the set of activated rules that support the results can be provided, discussed and argued by medical doctors, in the same way colleague's opinions are compared and this can stimulate knowledge base reviewing, in a constant process of refinement;
- lack of information, incoherent knowledge, uncertainty and hierarchical labeling can be handled;
- there is no need for training sets (like in neural network cases). Knowledge base can be built either manually, but this task is less time consuming, compared to setting up training sets (hundreds of CT scans with manual classification of volumes), or automatically from ontologies processing.

# 2. Background

## 2.1. CT scans

Computer Assisted Tomography (in short CT) scan [2] is a well established X-ray based acquisition procedure suitable for diagnoses of pathology and traumas as well as surgical preparation procedures. A patient's body image is acquired and rendered as a 3D volumetric information about X-ray absorption performed by various objects composing human body. Each *voxel* (volume element) of the scan has a typical size of 1 $\text{mm}^3$. Such volumetric information is arranged in a 3D matrix, where each cell contains a scalar number which corresponds to the radiometric absorption of that specific voxel (from now on *intensity*). The unit of measure is the HU (Hounsfield Unit). Different chemical elements, as well as different body structures, cause different X-ray absorptions. Therefore, 3D images can be used to identify the location of organs, vessels, bones, etc. and to assess their correct functional status. Note that different structures can show similar intensity levels. At the same time acquisition noise can reduce the clear perception of smaller issues. The goal of the training of radiologists is to cope with such issues and to be able to recognise finer details and problems. For example, typical tumours are less dense than healthy tissues and therefore they absorb less X-rays. This fact can be visualized on a 3D image as a darker area compared to the neighborhood, assuming to depict values with shades of gray ranging from black (low absorption, *e.g.*, air) to white (high absorption, *e.g.*, bones). However, while small tumours are more difficult to be identified, the prognosis is more favourable in case of early detection. Therefore computer assisted analysis have gained attention over the last decade, in order to improve the detection rate. There are various domains of analysis for a CT scan: *e.g.*, vision related ones are about organ and multi-organ segmentation (see [3] for a survey) and cancer detection (*e.g.*, [4]). Interestingly most methodologies, despite their sensible precision, are based on methods that are not xAI compliant (see Section 2.2). Our goal is therefore to design a novel pipeline that does not rely on such methods.

## 2.2. Explainable Artificial Intelligence

In the last decades we witnessed a very fast growing advances in AI based systems. In particular, from 2012 onward, sub-symbolic techniques such as Deep Learning (DL) became de-facto standards to deal with Computer Vision challenges, especially classification tasks such as recognition. Furthermore, they also showed great and increasing performances in object detection and image segmentation [5]. Another example of success is in natural language processing challenges with remarkable results in text generation [6]. This advance is also due to the accessibility of huge amount of information used to train systems, and by the availability of increasingly powerful hardware architectures that made it possible to use optimization algorithms on very large parameter spaces, such as those of deep neural networks. The measure of success of such techniques can be observed in how much AI is becoming more and more pervasive in daily life. Sub-symbolic methods heavily rely on learning algorithms over latent spaces that consist of billions of parameters and (in case of neural networks) hundreds of hidden layers, whose final goal is to learn approximations of an answer. Therefore, beside the impressing results obtained by DL, such approximations have been subject to criticism about their weakness to biasedness and the subsequent lack of robustness [7]. As another significant drawback of DL-based pipelines, is that such parametric spaces have to be treated as black-boxes, working as provider of intuitive information rather than transparent and reasoning-based decisions. For most situations the lack of transparency is not a problem, but their application to critical areas becomes an issue. In AI systems supporting medical decisions, the ability of interpret and explain the recommendations provided by the system is crucial. This also has to do with the concept of trustworthiness and reliability with reference to an *audience* of human users [8]. In recent years, the focus on ethical challenges in AI leads to move towards the so called *eXplainable Artificial Intelligence* (xAI), which aims at providing transparent and interpretable models and thus to increase trust in system based decisions. Explainability, and thus xAI, finds different definitions in literature, according to two main different approaches, namely the integration of inherently interpretable techniques in the system design, and the post-hoc explanation of what the system already have been done. The latter method is mostly employed in DL based AI, providing later description of the decision rather than unwrapping the black-box [9]. Such a method is yet under discussion, and concepts of explainability and causability in DL-based approaches are far from being attained.

In this work, the need of an intrinsically interpretable system is motivated by the degree of criticality of the task. As a matter of fact, it is desirable to introduce xAI techniques where the central point of the system regards the health of a patient and the detection of system errors of must be early and easy. The transparency allows, in fact, the users (Medical Doctors) to make informed choice and to be more confident when using the program. For this and other reasons, the topic of xAI in critical systems is being addressed by the European Commission, which developed an AI strategy to rule the trustworthiness of AI systems and enhance the excellence in such field [10, 11]. The development of such a strategy which led to a complex legal framework is motivated by the risk factors in the employment of AI-based systems in everyday life, as they became more and more pervasive as already said. In particular, one of the main reasons provided by the EU Commission is that whenever is not possible (or difficult) to find out why a decision or prediction has been made by an AI system, it consequently becomes not possible (or difficult) to detect whether someone has been unfairly disadvantaged. An AI improvement, according to the EU legal framework, must be subject to rules in order to protect the functioning of markets and the public sector, people's safety and fundamental rights. Such a legal framework divides AI systems in a pyramidal structure that identifies four degrees of risk, namely, minimal risk, limited risk, high risk and unacceptable risk. While unacceptable risk AI systems are clearly recognized as a threat for people's safety, and thus banned or prohibited, high risk systems comprises many AI technologies used in critical settings, such as health. Limited risk refers to systems which need specific transparency duties, such as chatbots, and finally, minimal risk are the vast majority of common systems, covering all applications that make everyday tasks more comfortable and/or easier. For the scope of this work, the proposed AI system falls in the high risk and thus subject to specific obligations before it can be proposed as a complete working system. Those obligations comprise (but are not limited to) the use of high quality of datasets as input of the system, in order to minimize discriminatory or riskful outcomes, the traceability of results, and the providing of clear and adequate information to the user. In particular, "High-risk AI systems shall be designed and developed in such a way to ensure that their operation is sufficiently transparent to enable users to interpret the system's output and use it appropriately." [11].

## 2.3. Human Anatomy Ontology

Clinical medicine uses standards for many types of data ranging from diseases (*International Classification of Diseases*[2] - ICD) to diagnoses and procedures (*Systematized Nomenclature of Medicine*[3] - SNOMED) and from laboratory data (*Logical Observation Identifiers Names and Codes*[4] - LOINC) to imaging data (*Digital Imaging and Communications in Medicine*[5] - DICOM). One of the earliest hierarchical terminology developed allows for cataloguing biomedical information and indexing journal articles (*Medical Subject Headings*[6] - MeSH). This controlled vocabulary is still used for MEDLINE, PubMed and National Library of Medicine databases. However, there are more formal and rich mechanisms to represent concepts and relationships, that is ontologies. In Computer Science, an ontology is a mean to formally model information and knowledge in a specific domain using simple representational primitives, such as classes, attributes, and relations among the members of the classes [12]. The concepts and relations belonging to the ontology are typically specified in a language that abstracts implementation strategies. In the literature, these languages are classified in various ways, however, they are commonly based on either first-order logic or description logic [13]. In particular, the Web Ontology Language (OWL) is a kind of language endorsed by the World Wide Web Consortium and conceived when the information needs to be processed by applications instead of just presented to humans [14]. In particular, OWL features include a collection of expressive operators for the concept, properties and relationships description, the ability to specify characteristics of properties (*e.g.*, transitivity, domains and ranges), and a semantic facilitating the use of inference and

---

[2] ICD: https://icd.who.int.

[3] SNOMED: https://www.snomed.org.

[4] LOINC: https://loinc.org.

[5] DICOM: https://www.dicomstandard.org.

[6] MeSH: https://www.nlm.nih.gov/mesh/meshhome.html.

reasoning [15]. The use of computational reasoning to answer complex questions has increased the development of ontologies in several medical fields supporting precision medicine and trustworthy artificial intelligence [16]. In biomedical and health sciences, ontologies are used to represent knowledge in different areas. The Disease Ontology extensively uses medical terminology standards (*e.g.*, MeSH, ICD, SNOMED) to semantically integrates disease and medical vocabularies [17], whereas the Infectious Disease Ontology is a set of interoperable ontologies covering the clinical aspects and the pathogens of most infectious diseases [18]. Other ontologies address the need for consistent descriptions of gene products across databases [19], represent phenotypic abnormalities encountered in human disease [20], annotate clinical trials [21], and support automated reasoning based on vaccine knowledge [22]. The Foundational Model of Anatomy ontology (FMA) represents a declarative knowledge about the whole human anatomy [23]. The last version of FMA (ver. 5.0.0) consists of 104,721 classes, organised hierarchically, and 168 properties that describe various characteristics of the concepts, such as "`constitutional_part_of`", "`adjacent_to`", "`dimension`" and "`bounds`", to name a few. For instance, the *Left lung* (code FMA:7310) is "`subclass_of`" *Lung* (code FMA:7195); is a "`constitutional_part_of`" *Left pulmopleural compartment* (code FMA:85056), *Left hemithorax* (code: FMA:20360) and *Intrathoracic part of chest* (code FMA:73438); and is related as "`regional_part_of`" to the *Lower lobe of left lung* (code FMA:7371) and *Upper lobe of left lung* (code FMA:7370). In this work, FMA codes of some high-level concepts are used to create the labels to be assigned to the cluster of voxels.

## 2.4. Constraint Satisfaction Problem

The field of automated reasoning requires ways of formalizing and describing scenarios and procedures with a great level of abstraction [24]. From this need stemmed the well-known *logic/declarative programming* paradigms. In contrast with the more common *imperative* and *object-oriented* paradigms, that require precise lists of instructions, the declarative ones "simply" need to state constraints and objective function [25]. This allows to describe a problem, and its solution, without the distraction of algorithmic details. Among various declarative paradigms, one of the most mature and employed in AI is *Constraint* Programming (CP) that, as said by [26], "is a powerful paradigm for solving combinatorial search problems". CP directly derives from the *constraint satisfaction problem*, that is the process of finding a solution through a set of constraints that impose conditions that the variables must satisfy [27]. Several techniques, that range from *search*, *constraint propagation*, and *backtracking*, are employed in solving CP models. Due to space limitations we will not provide further details on the topic and we address the interests readers to [26, 27] for a complete introduction on the topic of constraint programming.

In our work we utilize CP to identify sub-partitions of CT scans' regions (see Section 3.2). The problems that need to be solved are dynamically generated with constraints that depend on the analyzed regions. This can be easily modeled into constraint satisfaction problems, as we know the set of rules and constraints that a set of voxels must respect to be considered as a sub-region. Instead, trying to generate an imperative procedure flexible enough to tackle all the various cases could result impossible given the great variety of cases that different regions generate.

### 2.5. Answer Set Programming

Answer Set Programming (ASP) is a language that derives from the logic programming paradigm and that stems from the idea of *stable models* [28]. In particular, ASP can be used to model diverse domains through values, facts, variables, rules, and constraints. These domains are then solved trying to find a possible set of variables assignments, *i.e.*, a variable, if considered by the solution, must be assigned to a value without contradicting any constraint. More formally, as said by [29], a program $P$ in the language ASP is formed by set of rules $r$ of the form:

$$a_0 \leftarrow a_1, \ldots, a_m, not\ a_{m+1}, \ldots, not\ a_n$$

where $0 \leq m \leq n$ and each element $a_i$, with $0 \leq i \leq n$, is an *atom* of the form $p(t_1, \ldots, t_k)$, $p$ is a predicate symbol of arity $k$ and $t_1, \ldots, t_k$ are terms built using variables, constants and function symbols. Negation-as-failure (naf) literals are of the form $not\ a$, where $a$ is an atom. Let $r$ be a rule, we denote with $h(r) = a_0$ its *head*, and $B^+(r) = \{a_1, \ldots, a_m\}$ and $B^-(r) = \{a_{m+1}, \ldots, a_n\}$ the positive and negative parts of its *body*, respectively; we denote the body with $B(r) = \{a_1, \ldots, not\ a_n\}$. A rule is called a *fact* whenever $B(r) = \emptyset$; a rule is a *constraint* when its head is empty ($h(r) = \mathsf{false}$); if $m = n$ the rule is a *definite rule*. A *definite program* consists of only definite rules.

A term, atom, rule, or program is said to be *ground* if it does not contain variables. Given a program $P$, its *ground instance* is the set of all ground rules obtained by substituting all variables in each rule with ground terms. In what follows we assume atoms, rules and programs to be *grounded*. Let $M$ be a set of ground atoms ($\mathsf{false} \notin M$) and let $r$ be a rule: we say that $M \models r$ if $B^+(r) \nsubseteq M$ or $B^-(r) \cap M \neq \emptyset$ or $h(r) \in M$. $M$ is a *model* of $P$ if $M \models r$ for each $r \in P$. The *reduct* of a program $P$ w.r.t. $M$, denoted by $P^M$, is the definite program obtained from $P$ as follows: (i) for each $a \in M$, delete all the rules $r$ such that $a \in B^-(r)$, and (ii) remove all naf-literals in the the remaining rules. A set of atoms $M$ is an *answer set* [28] of a program $P$ if $M$ is the minimal model of $P^M$. A program $P$ is *consistent* if it admits an answer set.

The declarative nature of ASP allowed us to define a solid base containing the anatomical knowledge derived by ontologies or medical operators. We then used this knowledge base in our tool to autonomously classify the CT scan respecting the information provided by the experts. A more detailed explanation of how this process was envisioned and realized will be presented in Section 4.1.

## 3. Features extraction

### 3.1. Pre-processing and smoothing

One relevant issue about CT scan is that acquisition noise can be significant and it can even mask tumors. Radiologists are trained to mentally filter out noise and investigate scan content. Their first task is to correctly recognize the objects in the scan so they can then observe finer details, where texture (typical local arrangements of intensity absorption) can suggest potential pathologies. Often texture and noise merge and make the process difficult.

In this paper we focus on the first task and we set the general goal to interpret any voxel of the scan. This goes beyond the common goal in the literature where multi-organ detection targets main organs (liver, spleen, pancreas etc). Our approach could potentially identify any visible object according to a general ontology. Single organ detection precision, reached through DL, is still rather unsatisfactory, *e.g.*, up to 86% of pancreas volume is correctly classified [30]. We believe that the classification problem should be solved with a holistic approach, *i.e.*, an evaluation that considers at the same time every property about every element.

The processing pipeline starts with a rather strong smoothing of data. The resulting scan has little to no texture but it preserves boundaries between different regions. Organs and tissues are made of similar intensity voxels and any surface that separates such voxels is a suggestion to be considered during reasoning.

We selected a Total Variation based 3D smoothing named Directional Fast Gradient Projection [31]: this family of filters is particularly effective in CT scan domains and in surface preserving even in presence of strong noise. In Figure 1 we show in a rainbow palette the comparison between original raw data and smoothed version for a slice of a scan that intersects two vertebrae (red-blue color), a part of the kidney (top right dark green) and skin of the back (red bottom line). Cyan color is air. It can be noticed how structures are well preserved, while texture and noise are almost flattened.

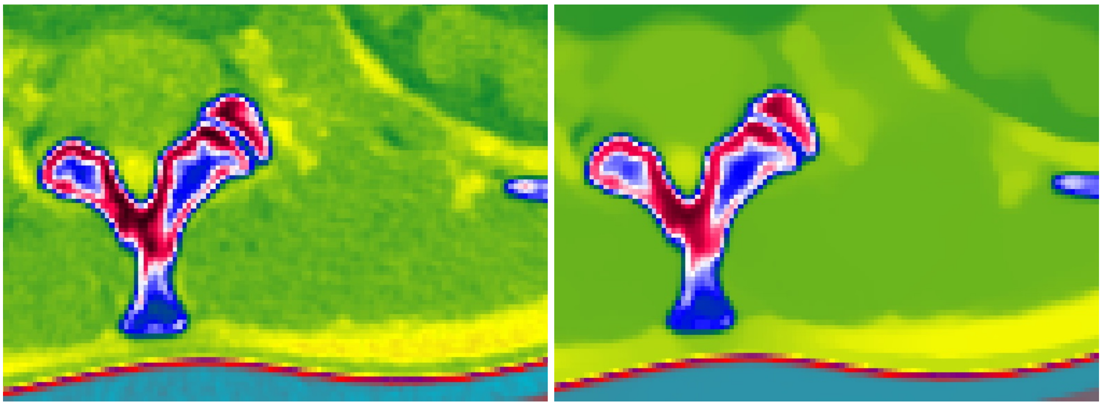

**Figure 1:** Raw data (left) and smoothed data (right)

The filtered version is processed with the goal of retrieving regular 3D regions. Unfortunately simple thresholding can not be applied, even if typical HU values are well documented: same intensity values can belong to different organs.

### 3.2. Geometric clustering

We set up a detection of regions that are defined in terms of simple mathematical relations. The resulting regions have little relation to target organs, but they can be handled as equivalence class for clustered voxels.

The idea is to group together voxels that have similar intensity and that are separated by sudden changes (high gradients). We show in Figure 2 on the left a simplification on a 2D domain. The same concepts can be generalized in the 3D space. Each point, associated to a voxel and a local intensity value, has a local neighbor (in the 3x3x3 window) with highest intensity (except for the local maximum). A graph induced by this relation defines connected components that reach a local maximum (dark blue and red points). This already represents a partitioning of the space, with the property that only monotonic paths are contained. A typical smooth CT scan, made of $512^3$ voxels may host some million local maxima. Inside a region, a finer analysis can reveal that along path there are maximal gradients values that cooperate to form a cut surface (see dashed line and black arrows that separate green and light blue points). It is important to separate such sub-partitions, since they represent visible changes that suggest an interface between different objects. In Figure 2 on the center, we depict with different colors the 3D points belonging to the same region but to different sub-regions: it can be seen that the bone-muscle interface (white - blue - green voxels, only one scan slice is depicted) is correctly partitioned by green and blue points (each point represents voxels in neighbor planes). Moreover the set of red points correctly suggests another division between white and blue voxels (different inner structures of the bone).

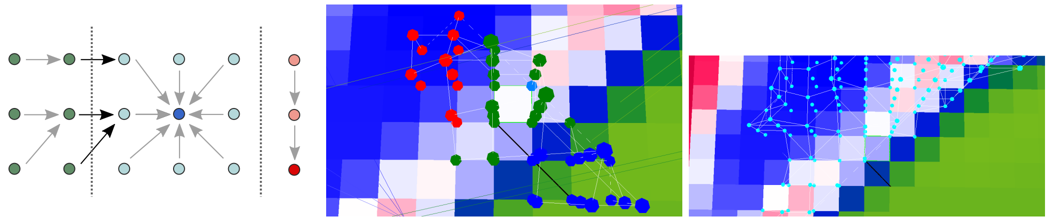

**Figure 2:** Example of geometric clustering. (left) Maximal gradient approximation for regions, (center) best surface cuts and sub-partitions, (right) clustering of sub-partitions.

In order to identify such sub-partitions inside a region, we set up a Minizinc [32] model that optimizes the best partitions that provide the highest cut in the corresponding graph. Each point is a variable that can be assigned to a partition code (3 different codes are enough for most cases). Each edge, defined above, is associated to the gradient magnitude. The constraints impose that the partition codes must satisfy the precedence imposed by edges directions. The only other constraint is that the local maximum is assigned to the maximum code (*i.e.*, 2 for our tests). The cost function is the sum of the edges that connect two different codes points. We found that Coin BC solver [33] was the most efficient among the ones provided by default Minizinc interface.

The resulting sub-partitions are almost constant valued, but their number is still very large to be used for reasoning. We also introduce a clustering that merges neighboring sub-partitions that also share same intensity and/or gradient directions along the shared surface of contact. Again, the resulting clustered voxels have similar intensities, but they collect a large set of sub-partitions. In Figure 2 on the right, we show with cyan points how the green sub-partition in the central picture gets merged with compatible neighboring partitions (in this case all red-white colored voxels).

The result of this procedure is a manageable set of (several thousands of) clusters that are retrieved by no medical information exploitation.

## 4. Ontology/Medical Knowledge Mapping into the Graph

Geometric clustering provides a set of partitioned volumes with annotated features that are considered from now on as the nodes of our search graph. Edges of such graph connect two volumes that share some surface of contact.

The goal of the labeling is to assign to each node a FMA code from the ontology. Since anatomic ontology is naturally hierarchical, we adapt our graph coloring like procedure in order to account for a hierarchical handling of FMA codes for domains. Ideally the root of the ontology represents any possible body's object, while following a particular branch in the ontology, it specializes the assignment for the specific node. Our optimal solution assigns the lowest label in the ontology to each node. Most quality results gets leaf values from the ontology.

Medical knowledge determines the constraints over ontology values and specify how spatial relations, intensity values and anatomic relations among objects apply.

We therefore define a set of ASP rules (the medical knowledge) and an ASP program that receives as input the graph structure as processed by the feature extraction.

### 4.1. ASP for Models Generation

As already mentioned, one of the main objectives of this work is to provide a tool that is able to support medical experts while providing proofs for its outputs. To accomplish this we decided to employ Answer Set Programming, through the Clingo solver [34], to analyze the information, expressed in the form of a graph, extrapolated by the scans. The motivation behind this choice is threefold: *i)* thanks to its declarative programming, ASP allows us to express anatomical rules in a concise and effective manner; *ii)* the process that leads ASP in finding the best model is completely traceable and only derived by the rules that reflect actual medical knowledge; and *iii)* finally, given the declarative nature of ASP and the consequent succinctness, enriching the program with new medical knowledge can be done fairly easily.

The main role that the ASP module of CARING undertakes is to identify the right class, w.r.t. the standard classifications introduced in Section 2.3, in which each node of graph falls into. This is done by inheriting the hierarchical approach typical of the medical knowledge.

In particular, we envisioned our ASP model to "firstly" identify the macro categories (the first level of FMA) of the various nodes. This means that each node is assigned to one, or more, of the following classes: `air`, `lung`, `fat`, `tissue`, `bone` or `unknown`. To do so we define the possible labels that each node can be assigned to by looking at their `intensity` values. While most of the nodes falls perfectly into only one category, some of them have values that can belong to multiple labels or even to none (that is why we also introduced the `unknown` category). The possible labels that a node `N` can be assigned to are captured by the set of ASP predicates `poss_lab(N,L)`, where `N` is the node id and `L` is one of the aforementioned labels. This predicate is defined as follows:

$$\begin{aligned}\texttt{poss\_lab(N,L):-}\quad & \texttt{node(N,I)},\texttt{lab(L)},\texttt{in\_min(L,MIN)},\texttt{in\_max(L,MAX)},\\ & \texttt{in\_tolerance(L,T)},\texttt{I}+\texttt{T}\geq\texttt{MIN},\texttt{I}-\texttt{T}<\texttt{MAX}.\end{aligned}$$

where: *i)* `node(N,I)` is the description of a node, in terms of id and intensity, provided by the features extraction procedure; *ii)* `lab(L)` express that `L` must belong to the aforementioned set of labels; *iii)* `in_min(L, MIN)` and `in_max(L, MAX)` describe the range in which a label `L` is defined by providing its minimum and maximum intensity, respectively; and *iv)* `in_tolerance(L,T)` provides the tolerance `T` of the intensity range for the label `L`.

After generating the possible label domain for each node we try to minimize the number of unknown. This is done by allowing the unknown nodes to assume their neighbors' labels if their intensity is not too distant. The ASP model will return, as solution, the one that minimize the number of unknown nodes. An example of one of the rules used to reduce the unknown is as follows:

```
poss_lab(N, L):-   poss_lab(N, unknown), node(N, I), lab(N2, L), L! = unknown,
                   edge(N, N2, _, _), in_tolerance(L, T1), T = T1 * 2,
                   in_min(L, MIN), in_max(L, MAX), I + T ≥ MIN, I − T ≤ MAX.
```

where: *i)* `lab(N2,L)` is the label assigned to `N2` in the solution of the model, generated by "`1{lab(N, L) : poss_lab(N, L)}1:-node(N, _).`"; and *ii)* `edge(N,N2,_,_)` is the fact, derived by the imperative procedure, that indicates that nodes `N` and `N2` are connected in the graph.

Finally, the ASP program also discriminates among different entities that fall in the same category. In particular, for this work, we focused on distinguishing the various bones that were present in our working area. To do so we defined some rules that relate intensity of the nodes, labelled as bones, in relation to the minimal and maximal intensity of the contact surface with other bone-labelled nodes. This allowed us to identify bone-labelled nodes with a strong difference between the average intensity and the intensity on the contact surface, hinting that the two connected nodes may belong to two different bones. This is because two bones that are in contact are always separated by lower intensity zones. We then impose that the nodes that belong to the same bone must be transitively connected. Some results of this classification process are presented in Figure 3.

This is a minimal test-bed domain that we used to show our the resolution schema rather than the overall performances. The accuracy of the program, in fact, is under constant improvement as it is extended to cover additional regions and sub-labels.

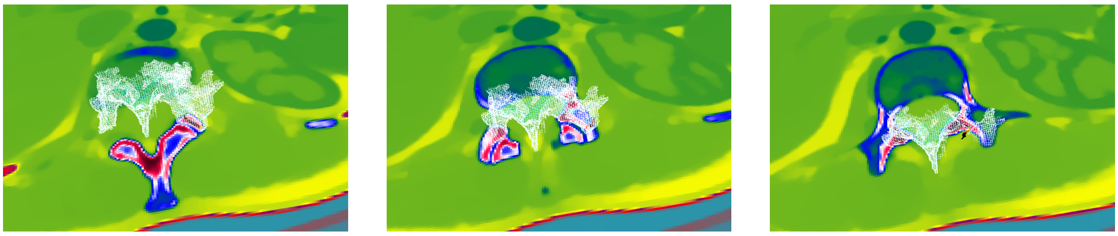

**Figure 3:** An example of result of the CARING procedure. The tool allows to extract the set of voxels that describe specific anatomical objects. In this case we rendered (with a white surface) the set of point that CARING assigned to a single vertebra. From left to right we see the vertebra rendered on different slices of the scan.

Let us note that, even if our description of the ASP model is presented as a sequence of properties, the solving process of ASP takes advantage of all conditions simultaneously. We refer the interested reader to [35] for an insightful explanation of this solving process.

# 5. Results

Our proof of concept processed a raw abdominal CT scan with size 512x512x70 voxels. Total variation filtering on a P100 Nvidia GPU took a few seconds of processing. We focused on a region with 110 x 110 x 38 voxels that covers two vertebrae, one rib, the lower end of a lung, part of a kidney and spleen.

Geometric clustering for the whole scan produced 740K maximal voxels (average of 25 voxels per partition), while the region of interest contained 5K maximal voxels (average of 83 voxels per partition). Computing sub-partitions with Minizinc (5K CSPs solved in a batch task) took 80 minutes on a 2.3GHz i9 Macbook Pro. This phase will require some optimization/reencoding in order to make it scalable for complete scans (100x–1000x of speedup needed).

Minizinc computation returned 22K sub-partitions, that were clustered into 5355 clusters. Based on intensity ranges, we directly labelled 3931 clusters with most general FMA code. The ASP procedure labelled 1424 clusters with a time of 0.602 seconds. The procedure is still in its infancy, as the accumulated medical knowledge is far from being complete. Nonetheless CARING is able to discriminate between the macro categories of anatomical bodies and to label geometric features into compatible clusters. Our proof of concept implements the complete xAI pipeline and shows the feasibility of the approach. Our current knowledge base is limited but we expect promising results with a more extended set of properties.

# 6. Conclusions

In this paper we described the design of an xAI pipeline for the interpretation of CT scans. We covered the foundational aspects that helped us to guarantee explainability of the deduction process. As proof of concept, we implemented a first model that was able to analyze a region of a CT scan and to recover correct classification for contained regions.

This work can be extended along several directions: *i)* quality and efficiency improvement for the feature extraction phase; *ii)* increase the scalability of the ASP reasoner to tackle complete CT scans; *iii)* a multi level and hierarchical search for labeling ontology codes; *iv)* the definition of a knowledge base automatically extracted from FMA codes and their relationships; *v)* the possibility to manually add knowledge, with handling of liability and handling of inconsistent knowledge; *vi)* generate natural language descriptions for Medical Doctors' validation; *vii)* testing over challenging scans; and *viii)* comparison with the state of the art non-xAI systems.

# Acknowledgments

We would like to thank Medical Doctors Mirko D'Onofrio and Nicolò Cardobi (University of Verona) for providing CT scan data and useful discussions.

## References

[1] F. Wacker, H. Lippert, R. Pabst, Arterial variations in humans: Key reference for radiologists and surgeons, Thieme (2017).

[2] G. T. Herman, Fundamentals of computerized tomography: image reconstruction from projections, Springer Science & Business Media, 2009.

[3] M. Kardell, Automatic segmentation of tissues in ct images of the pelvic region. master thesis, 2014.

[4] L. C. Chu, et al., Application of deep learning to pancreatic cancer detection: Lessons learned from our initial experience, Journal of the American College of Radiology 16 (2019) 1338–1342.

[5] A. Krizhevsky, et al., Imagenet classification with deep convolutional neural networks, Advances in neural information processing systems 25 (2012).

[6] T. Brown, et al., Language models are few-shot learners, Advances in neural information processing systems 33 (2020) 1877–1901.

[7] G. Marcus, Deep learning: A critical appraisal, arXiv preprint arXiv:1801.00631 (2018).

[8] A. B. Arrieta, et al., Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai, Inf. fusion 58 (2020) 82–115.

[9] A. Holzinger, et al., Causability and explainability of artificial intelligence in medicine, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 9 (2019).

[10] E. Commission, A european approach to artificial intelligence, https://digital-strategy.ec.europa.eu/en/policies/european-approach-artificial-intelligence, 2022.

[11] E. Commission, Proposal for a regulation of the european parliament and of the council laying down harmonised rules on artificial intelligence (artificial intelligence act) and amending certain union legislative acts, https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:52021PC0206&from=EN, 2021.

[12] N. Guarino, D. Oberle, S. Staab, What is an ontology?, in: Handbook on ontologies, Springer, 2009, pp. 1–17.

[13] J. Pulido, M. Ruiz, R. Herrera, E. Cabello, S. Legrand, D. Elliman, Ontology languages for the semantic web: A never completely updated review, Knowledge-Based Systems 19 (2006) 489–497.

[14] D. L. McGuinness, F. Van Harmelen, et al., Owl web ontology language overview, W3C recommendation 10 (2004) 2004.

[15] S. Bechhofer, M. T. Özsu, L. Liu, Owl: Web ontology language, in: {Encyclopedia of Database Systems}, Springer Nature, 2009.

[16] M. A. Haendel, C. G. Chute, P. N. Robinson, Classification, ontology, and precision medicine, New England Journal of Medicine 379 (2018) 1452–1462.

[17] L. M. Schriml, et al., Human disease ontology 2018 update: classification, content and workflow expansion, Nucleic acids research 47 (2019) D955–D962.

[18] A. Ruttenberg, A. Goldfain, A. Diehl, B. Smith, B. Peters, Infectious disease ontology, The National Center for Biomedical Ontology 5 (2016).

[19] M. Ashburner, et al., Gene ontology: tool for the unification of biology, Nature genetics 25 (2000) 25–29.

[20] S. Köhler, et al., The human phenotype ontology in 2021, Nucleic acids research 49 (2021)

D1207–D1217.

[21] B. Smith, et al., The obo foundry: coordinated evolution of ontologies to support biomedical data integration, Nature biotechnology 25 (2007) 1251–1255.

[22] Y. Lin, Y. He, Ontology representation and analysis of vaccine formulation and administration and their effects on vaccine immune responses, Journal of biomedical semantics 3 (2012) 1–15.

[23] N. F. Noy, M. A. Musen, J. L. Mejino Jr, C. Rosse, Pushing the envelope: challenges in a frame-based representation of human anatomy, Data & Knowledge Engineering 48 (2004) 335–359.

[24] O. Torgersson, A note on declarative programming paradigms and the future of definitional programming, in: In Proceedings of Das Wintermote 96, 1996.

[25] A. Bockmayr, J. N. Hooker, Constraint programming, in: K. Aardal, G. Nemhauser, R. Weismantel (Eds.), Discrete Optimization, volume 12 of *Handbooks in Operations Research and Management Science*, Elsevier, 2005, pp. 559–600. URL: https://www.sciencedirect.com/science/article/pii/S0927050705120106. doi:`https://doi.org/10.1016/S0927-0507(05)12010-6`.

[26] F. Rossi, P. van Beek, T. Walsh, Handbook of Constraint Programming, Elsevier Science Inc., USA, 2006.

[27] E. P. K. Tsang, Foundations of constraint satisfaction., Computation in cognitive science, Academic Press, 1993.

[28] M. Gelfond, V. Lifschitz, The stable model semantics for logic programming., in: ICLP/SLP, volume 88, 1988, pp. 1070–1080.

[29] F. Fabiano, A. Dal Palù, An ASP approach for arteries classification in CT scans, J. Log. Comput. 32 (2022) 331–346.

[30] Y. Zhou, L. Xie, W. Shen, E. Fishman, A. Yuille, Pancreas segmentation in abdominal ct scan: a coarse-to-fine approach, arXiv preprint arXiv:1612.08230 (2016).

[31] M. J. Ehrhardt, M. M. Betcke, Multicontrast mri reconstruction with structure-guided total variation, SIAM Journal on Imaging Sciences 9 (2016) 1084–1106.

[32] N. Nethercote, P. J. Stuckey, R. Becket, S. Brand, G. J. Duck, G. Tack, Minizinc: Towards a standard cp modelling language, in: C. Bessière (Ed.), Principles and Practice of Constraint Programming – CP 2007, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 529–543.

[33] J. Forrest, et al., coin-or/cbc: Release releases/2.10.8, 2022. URL: https://doi.org/10.5281/zenodo.6522795.

[34] M. Gebser, R. Kaminski, B. Kaufmann, T. Schaub, Multi-shot ASP solving with clingo, CoRR abs/1705.09811 (2017).

[35] M. Gebser, R. Kaminski, B. Kaufmann, T. Schaub, Answer Set Solving in Practice, Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan & Claypool Publishers, 2012.