

Article

Not peer-reviewed version

Machine Learning Platform for Disease Diagnosis with Contrast CT Scans

[Jennifer Jin](#) , [Mira Kim](#) , [Soo Dong Kim](#) ^{*} , Daniel Jin

Posted Date: 29 July 2024

doi: 10.20944/preprints202407.2284.v1

Keywords: machine learning platform; medical image analytics; contrast-enhanced ct; open closed principle (OCP); liver cancer diagnosis



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Article

Machine Learning Platform for Disease Diagnosis with Contrast CT Scans

Jennifer Jin ¹, Mira Kim ², Soo Dong Kim ^{3,*}, and Daniel Jin ⁴

¹ School of Computer Science and Engineering, California State University, San Bernardino, CA 92407, USA

² School of Computer Science, California State University, Fullerton, CA 92831, USA

³ School of Software, Soongsil University, Seoul, 06978, Korea

⁴ Department of Radiology, Loma Linda University Medical Center, Loma Linda, CA 92354, USA

* Correspondence: sdkim777@gmail.com; Tel.: +1-949-550-8225

Abstract: Machine learning has gained significant recognition within healthcare organizations as a potent technology for medical image analytics. Among various medical image modalities, Contrast-Enhanced CT (CECT) is widely utilized to acquire additional diagnostic information that can help to better visualize and evaluate certain structures or abnormalities in the human body. Accordingly, there is an increasing demand for machine learning systems for disease analytics using CECT. However, developing such systems presents two significant challenges: high technical complexity and high development effort. This paper presents a software platform that can effectively remedy these challenges. This platform embodies a unified software process for CECT-specific disease diagnosis including oncological diseases, cardiovascular diseases, and gastrointestinal diseases. It integrates multiple types of machine learning models into the disease diagnosis process, where the models can diagnose lesions, malignancies, tumors, medical features of tumors, temporal transitions with a contrast agent, and disease progress. The platform has been implemented in Python using Scikit-learn and TensorFlow libraries. To validate its applicability and reusability, a Hepatocellular Carcinoma (HCC) diagnosis system has been implemented.

Keywords: machine learning platform; medical image analytics; contrast-enhanced ct; open closed principle (OCP); liver cancer diagnosis

1. Introduction

Machine learning (ML) has gained significant recognition within healthcare organizations as a potent tool for medical image analytics, particularly in the realm of disease diagnosis. More specifically, ML models can effectively extract meaningful insights from medical images, identify tumor occurrences, stage tumor progression, determine disease occurrences, predict future disease occurrences, and even recommend treatment procedures. Some of the commonly used ML algorithms include Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Generative Adversarial Networks (GAN), and Support Vector Machine (SVM).

Among various medical image modalities, Contrast-Enhanced CT (CECT) is widely utilized for disease assessment since CECT offers several advantages over non-contrast CT. The advantages include improved visibility of blood vessels, better characterization of lesions, enhanced detection of tumors, improved assessment of organ perfusion, clearer delineation of anatomical structures, and detection of active inflammation. Some of the diseases that can be better diagnosed using CECT include oncological diseases, cardiovascular diseases, and gastrointestinal diseases.

However, developing ML-based diagnosis systems with CECT presents two significant challenges: substantial technical complexity and high development effort. Consequently, many of these systems suffer from limited diagnostic performance and are not accepted for use in medical clinics.

High technical complexity of developing these systems is attributed to transforming the diagnosis processes applied by physicians into software processes. It is challenging mainly due to the deviations between the terms, practices, and standards used by physicians and those of software developers. Moreover, it is challenging to configure most suitable ML algorithms for various diagnostic tasks, training with optimal set of hyperparameters, fine-tuning trained ML models for optimal performance, managing control flows and data flows among the ML models, and synthesizing various diagnostic results to determine ongoing diseases and their progression.

High development effort is attributed to the considerable time required for designing the system architecture, functional components, data persistence, and runtime behavior. Additionally, the effort needed for designing, training, and tuning multiple ML models trained with different algorithms can be significantly high. This process also requires a team of skilled professionals, including data scientists, software engineers, and medical domain experts.

Our strategy to address these challenges is to develop a machine learning-based software platform that captures the common diagnostic process with CECT, provides an essential set of functional components, manages data persistence, and supports a high level of configurability and extensibility for developing various CECT-based diagnostic systems. More specifically, we designed the platform by applying the Open-Closed Principle, where common features are designed to be fixed, and variable features among diagnostic systems are designed to be customizable and extendable. The significant advantages of leveraging this platform for developing CECT-based diagnostic systems are the substantial reduction in technical complexity and the cost-effective development of high-quality diagnostic solutions.

The paper is organized as follows: Section 2 summarizes related works to medical image analytics. Section 3 provides the fundamentals of medical analytics with CECT. Section 4 presents the design of the platform in terms of platform architecture, components, diagnostic process, database, and machine learning models. Section 5 elaborates on a case study of developing an HCC diagnosis system using the proposed platform.

2. Related Works

This section is to summarize representative works on machine learning-based disease diagnosis with CT scan.

Tao et al. [1] presents a case report of a renal anastomosing hemangioma, an uncommon variant that histologically resembles angiosarcoma. The authors describe the clinical and pathological features of this rare lesion and provide a literature review to distinguish it from angiosarcoma for accurate diagnosis. Wang et al. [2] describe the development and deployment of an AI-assisted CT imaging analysis system for COVID-19 screening. The system leverages deep learning algorithms to analyze CT images, providing accurate and efficient COVID-19 detection. The study demonstrates the system's effectiveness in improving screening workflows during the pandemic.

Kouanou et al. [3] use the Hadoop framework and Spark framework to develop appropriate methods to handle large image data sets for classification. Mansour et al. [4] present an Artificial Intelligence (AI) and Big Data Analytics-based Intracerebral Hemorrhage e-diagnosis (AIBDA-ICH) model using CT images. The presented model utilizes IoMT devices for data acquisition process, involving graph cut-based segmentation model for identifying the affected regions in the CT images. Dhar et al. [5] traverses the major challenges that the deep learning community in medical image diagnosis, like the unavailability of balanced annotated medical image data, adversarial attacks faced by deep neural networks and architectures due to noisy medical image data, a lack of trustability among users and patients, and ethical and privacy issues related to medical data.

Xie et al. [6] conduct design activities to formulate a system that enables physicians to explore and understand AI-enabled chest X-ray analysis using a paired survey between referring physicians and radiologists reveals whether, when, and what kinds of explanations are needed, a low-fidelity prototype co-designed with three physicians formulates eight key features. Olveres et al. [7] cover different AI advances that tackle medical problems such as cardiology, cancer, dermatology, neurodegenerative disorders, respiratory problems, and gastroenterology. Different imaging is

utilized to build automatic systems that help medical diagnosis, with limitations due to signal to noise ratio, contrast and resolutions in time, space and wavelength.

Lei et al. [8] introduce the background of AI and its application in breast medical imaging (mammography, ultrasound and MRI), such as in the identification, segmentation and classification of lesions; breast density assessment; and breast cancer risk assessment. Lutnick et al. [9] create an intuitive interface for data annotation and the display of neural network predictions within a commonly used digital pathology whole-slide viewer. They demonstrate that segmentation of human and mouse renal micro compartments is repeatedly improved when humans interact with automatically generated annotations throughout the training process. Mehta et al. [10] conduct a systematic mapping study to identify and analyze research on big data analytics and artificial intelligence in healthcare with focus on application of analytics, machine learning and artificial intelligence over big data enables identification of patterns and correlations and hence provides actionable insights.

Ker et al. [11] introduce the machine learning algorithms as applied to medical image analysis, focusing on convolutional neural networks, and emphasizing clinical aspects of the field. Chen et al. [12] propose a convolutional autoencoder deep learning framework to support unsupervised image features learning for lung nodule through unlabeled data, which only needs a small amount of labeled data for efficient feature learning. Willemink et al. [13] describe fundamental steps for preparing medical imaging data in AI algorithm development, explain current limitations to data curation, and explore new approaches to address the problem of data availability. Shi et al. [14] review AI techniques in Imaging Data Acquisition, Segmentation and Diagnosis for COVID-19.

Contrast enhanced computed tomography (CT) is a standard of care for the radiological diagnosis of various types of diseases including Circulatory system disease, kidney and bladder stones, inflammatory diseases, heart disease and liver masses. Gosalia et al. [15] evaluate the ability of contrast-enhanced CT to detect acute myocardial infarction (MI), which has not been systematically assessed. On contrast-enhanced helical chest CT, we retrospectively identified 18 patients with an initial MI. They conclude Acute MI is detectable on contrast-enhanced chest CT as an area of decreased left ventricular myocardial enhancement in a specific coronary arterial distribution. Gonio et al. [16] investigate the radiological findings prognostic for the development of pancreatic adenocarcinoma in a cohort of patients with hepatocellular carcinoma, using multiphasic CT.

Chang et al. [17] evaluates the accuracy of contrast-enhanced ultrasound (CEUS), which is safe for patients with chronic kidney disease, for the characterization of kidney lesions in patients with and without chronic kidney disease. Cayet et al. [18] assess CT-scan performance for Sinusoidal obstruction syndrome (SOS) diagnosis for patients receiving neoadjuvant chemotherapy (NC) prior to CRLM surgery, comparing obtained results with pathological gold standard. Rawson and Pelletier [19] detail the types of contrast agents, and describe when to order Contrast-Enhanced CT.

Ohno et al. [20] explore the basics of dual-energy CT and dynamic first-pass contrast-enhanced perfusion CT techniques, the basics of time-resolved contrast-enhanced MRA and dynamic first-pass contrast-enhanced perfusion MRI, and clinical applications of contrast-enhanced CT- and MRI-based perfusion assessment for patients with pulmonary nodule, lung cancer, and pulmonary vascular diseases. Scheffel et al. [21] assess the value of dual-energy CECT imaging for the detection of urinary stone disease using dual-source CT and suggest Virtual unenhanced CT images reconstructed from contrast-enhanced dual-energy CT allow detection of urinary stones with good sensitivity and excellent specificity, but sensitivity is decreased in abdominal obese patients. Fletcher et al. [22] demonstrate contrast-enhanced CT colonography is a promising method for detecting local recurrence, metachronous disease, and distant metastases in patients with prior invasive colorectal carcinoma.

Lavarone et al. [23] suggest Intrahepatic Cholangiocarcinoma (ICC) which may develop in patients with cirrhosis, in cirrhotic patients displays distinct vascular patterns at CT-scan that allow for differentiation from Hepatocellular Carcinoma (HCC). Chen et al. [24] compare the enhancement pattern of ICC on contrast-enhanced ultrasound (CEUS) with that on CECT and conclude the

enhancement patterns of ICC on CEUS were consistent with those on CECT in the arterial phase, whereas in the portal phase ICC faded out more obviously on CEUS than on CECT. Prionas et al. [25] suggest conspicuity of malignant breast lesions, including ductal carcinoma in situ, is significantly improved at contrast-enhanced breast CT and quantifying lesion enhancement may aid in the detection and diagnosis of breast cancer. Vuppala et al. [27] describe a framework for leveraging the cloud platform and big data technologies to perform medical image data analytics even across multiple institutions, addressing security, privacy and scalability using a newer generation de-identification technique based on the big data platform.

Chrimes et al. [28] established a Healthcare big data analytics (HBDA) platform and tested its performance for different patient query types to demonstrate high usability for a variety of reporting requirements by providers and health professionals. Allen et al. [29] demonstrate a robust platform that uses software automation and high-performance computing (HPC) resources to achieve real-time analytics of clinical data, specifically magnetic resonance imaging (MRI) data. Kaur et al. [30] propose a generic architecture for enabling AI based healthcare analytics Platform by using open sources and present the importance of applying AI based predictive and prescriptive analytics techniques in Health sector. The system will be able to extract useful knowledge that helps in decision making and medical monitoring in real-time through an intelligent process analysis and big data processing.

The current work on medical image analytics with CT scans primarily focuses on addressing software approaches for diagnosing diseases using medical images. Some works have been developed to utilize machine learning models for segmentation and classification tasks involved in the diagnostic processes. A few works propose platform approaches to disease diagnosis that are not tied to specific medical image schemes or machine learning model configurations.

Our work is uniquely distinguished from existing works in three aspects: (1) defining a unified software process to automate disease diagnosis with CECT scans, (2) configuring specific ML models for various diagnostic tasks, and (3) designing core components of the platform by applying the Open-Closed Principle (OCP) [33] to provide high configurability for key variability areas.

3. Foundation of Medical Image Analytics with Machine Learning

This section provides a foundation for medical image analytics with ML models and disease diagnosis with CECT.

3.1. Machine Learning Algorithms for Medical Image Analytics

Medical Imaging Analytics refers to the application of software analytical methods to medical imaging data to extract useful medical information including health condition and on-going disease. ML plays a key role in developing high performing medical image analytics systems. The essential types of ML algorithms for medical image analytics include the following.

- CNN for image classification, segmentation, and object detection tasks
- RNN for analyzing time-series medical images or volumetric data
- GAN for image data augmentation, image synthesis, and image-to-image translation
- SVM for medical image classification and abnormality detection

There exist multiple ML algorithms for each type of ML model. For example, representative CNN algorithms that can be used for medical image analytics include U-Net, V-Net, DeepMedic, ResNet, Inception, and Mask R-CNN. The most optimal ML algorithm for a given diagnostic task should be identified and applied to yield high-performance analytics.

3.2. Contrast-Enhanced Computed Tomography

CECT scan is a medical imaging technique that uses CT imaging in conjunction with a contrast agent to produce detailed images of internal organs, blood vessels, and other tissues.

A contrast Agent is a substance, typically iodine-based or barium-sulfate-based, that is administered to the patient either orally, intravenously, or rectally before or during the CT scan. The

contrast agent increases the contrast within the images by highlighting specific areas, thereby improving the visibility of organs, tissues, and vessels.

CECT offers significant advantages over Non-contrast CT, particularly for specific diagnostic tasks. A primary benefit is the enhanced visualization of target objects in CT slices, which is particularly significant for several types of anatomical and pathological structures. For example, blood vessels and vascular structures such as arteries and veins are much more clearly defined with the use of contrast agents.

Another key benefit of using CECT-based diagnosis is the ability to observe the temporal evolution of medical conditions. This includes monitoring changes in the size, shape, and characteristics of lesions, tumors, or other abnormalities over time. This information is crucial for assessing the progression or regression of diseases, evaluating the effectiveness of treatments, and making informed clinical decisions. An example of CT slices showing temporal changes of a lesion is shown in Figure 1.

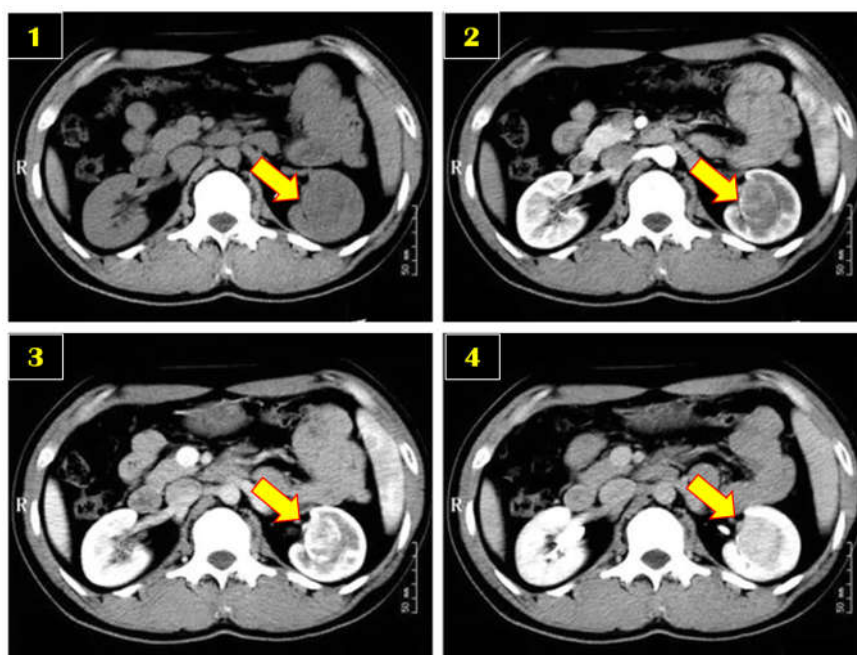


Figure 1. Temporal Changes of Lesion with Contrast Enhanced CT.

The CT image in the figure shows axial slices of the abdomen, specifically highlighting the liver. The arrows indicate lesions of interest within the liver. The figure also reveals the temporal changes of the organ over four phases. The arrows point to a lesion in the liver, showing the progression or response of a hepatic lesion over different phases of contrast enhancement.

The CT slice #1 shows the liver in the unenhanced (non-contrast) phase, captured in an axial plane. CT slice #2 depicts the liver in the arterial phase after the administration of contrast material. CT slice #3 represents the liver in the venous phase, and CT slice #4 displays the liver in the delayed phase. This observation of temporal evolution on medical objects is essential in diagnosing some diseases such as Liver Cancer [31].

3.3. Tasks in Disease Diagnosis Process with CECT

Disease diagnosis with CECT involves a number of common tasks to comprehensively assess the presence, extent, and progression of diseases.

Not all diagnostic tasks should be performed with ML models. Tasks with low complexity such as the characterization of lesions and tumors by size, shape, and density do not necessarily require ML models. Many of the complex diagnostic tasks can be performed more accurately with ML models

for segmentation and classification purposes, and such diagnostic tasks are summarized are summarized in Table 1.

Table 1. Diagnosis Tasks Performed by Machine Learning Models.

Diagnosis Tasks	Medical Objects
Segmentation	Organ
	Lesion
	Tumor
Classification	Medical Features of Tumors
	Temporal Change of Tumors
	Occurrence of Diseases
	Stage of Diseases

This set of diagnosis tasks becomes the basis for designing ML models and deriving functional components of the platform.

4. Design of the Platform

This section elaborates the detailed design of the proposed platform in terms of the platform architecture, functional components, disease diagnosis process, database, and ML algorithms.

A software platform is not a specific application, rather, it provides a set of functions and features that are common among potential applications constructed with the platform. Therefore, the reusability of the platform and the configurability for specific applications are key quality requirements. Our approach to meet these requirements is to design the platform by applying Open-Closed Principle (OCP) [33]. In this design principle, the common features of a system are designed to be closed from modifications and the variable features of the system are designed to be open for extension and customization.

4.1. Architecture Design with Micro Kernal Style

The architecture of a software system defines the stable and schematic layout of the system, where various components are deployed. The architecture of software platforms reveals a higher degree of variability than the architecture of a single application. This is because there are various types of differences, i.e., variation points, among application systems that are built upon the platform [34]

This becomes especially evident with the medical analytics platform proposed in this paper, where variability exists on the target disease types, diagnosis processes, machine learning models deployed, the policy for determining disease occurrences and their stages.

To handle the variability, we applied the Microkernel architecture style in designing the platform architecture as shown in Figure 2.

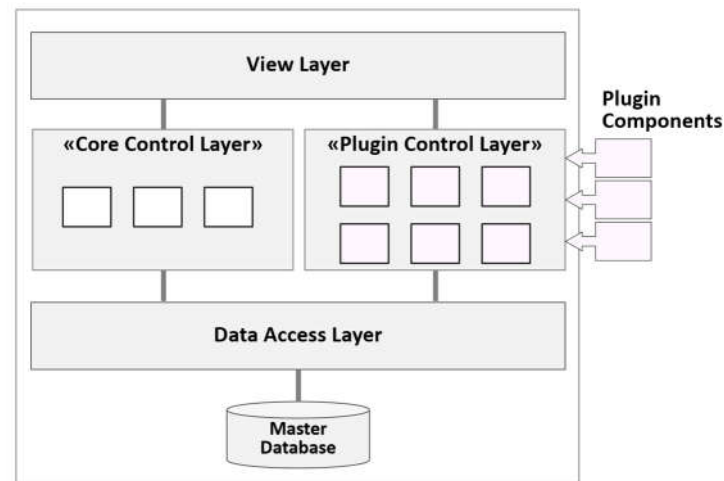


Figure 2. Schematic Architecture with Micro Kernel style.

The Microkernel architecture style is characterized by the division of system responsibilities between the Core Control layer and the Plugin Control layer, which binds externally devised plug-in components on-the-fly. The Core Control layer provides functionality that is common among various diagnosis systems and is designed with closed-design schemes.

The Plugin Control layer defines a set of Required Interfaces in UML, which are implemented/realized in plugin components. Externally implemented plugin components are then bound to the Plugin Control layer to fulfill the variation points.

Benefits of adopting Microkernel architecture style in building this platform are as follows:

- **Separation of Concerns:** Microkernels separate the invariant components from components with variability, allowing for a clear separation of concerns.
- **Customizability:** The functionality of components with variability can be customized for specific diseases, diagnosis methods, and ML models applied. Updating or replacing plug-in objects does not require modifying the kernel, making the system adaptable and extensible.
- **Platform Independence:** The components in the Core Control Layer are stable and platform-specific elements, making it easier to port to different medical image analytics applications.

4.2. Design of Functional Components

To derive functional components effectively, we first constructed a use case model that captures the platform functionality by applying the commonality and variability analysis [] on a set of representative family applications, including Liver Imaging Reporting and Data System (LI-RADS) [31], Lung Imaging Reporting and Data System (LUNG-RADS) [32], Breast Imaging Reporting and Data System (BI-RADS) [38], and Prostate Imaging Reporting and Data System (PI-RADS) [39] applications. For this use case diagram, we derived 12 functional components by clustering related use cases into a component.

A key feature of any software platform is its capability of enabling effective customizing of platform functionality. For this reason, these functional components were analyzed for their potential variability. Functional components with variability are placed in the Plugin Control layer, and components with no variability are placed in Core Control layer, as shown in Figure 3.

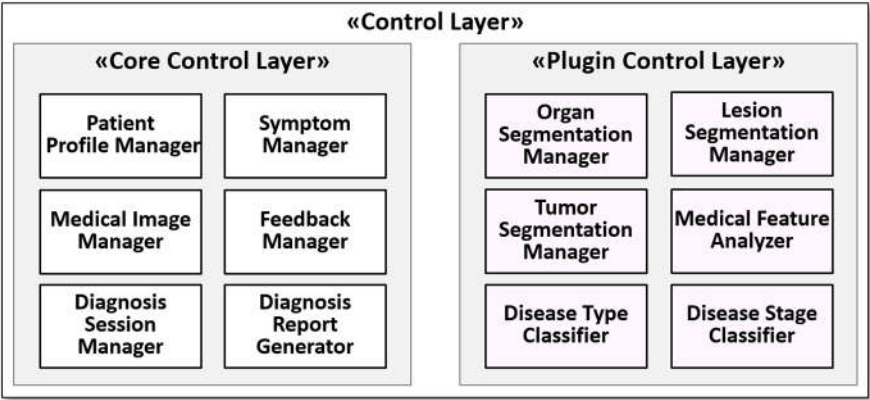


Figure 3. Functional Components of the Platform.

Among the identified components, the six components in the Plugin Control layer were determined to have their intrinsic variability.

The design of functional components with variability can effectively be done by applying Commonality and Variability (C&V) Analysis [34]. The result of applying variability modeling on these components is summarized in Table 2.

Table 2. Variability Analysis of the Functional Components.

Functional Components	Variability Type	Variation Points	Variant Scope	Set of Variants
Organ Segmentation Manager	Attribute	Organ	Selection	Organ of Interest for Target Disease
	ML Model	CNN Segmentation Model	Open	U-Net, V-Net, RCNN, FCNN, SegNet, etc.
Lesion Segmentation Manager	Attribute	Set of Target Lesions	Selection	Types of Lesions for Target Disease
	ML Model	CNN Segmentation Model	Open	U-Net, V-Net, RCNN, FCNN, SegNet, etc.
Tumor Segmentation Manager	Attribute	Set of Target Tumors	Open	Types of Tumors for Target Disease
	ML Model	CNN Segmentation Model	Open	U-Net, V-Net, RCNN, FCNN, SegNet, etc.
	ML Model	RNN Classification Model	Open	Basic RNN, LSTM, GRN, Convolutional LSTM, etc.
Medical Feature Analyzer	Attribute	Set of Medical Feature Types	Open	Types of Medical Features for Target Disease
	ML Model	Set of ML Models	Open	ML Models for Analyzing Target Medical Features
Disease Type Classifier	Logic	Algorithm for Disease Classification	Open	Algorithm for Classifying Occurrences of Target Disease
Disease Stage Classifier	Logic	Algorithm for Stage Classification	Open	Algorithm for Classifying Stage of Disease Occurrence

The variability analysis becomes the basis for deriving design schemes to effectively customize the variability, such as defining required interfaces for ‘Open’-scoped variation points.

For example, the *Tumor Segmentation Manager* for a specific organ can be defined with a required interface, which can bind different implementations of a CNN segmentation, as shown in Figure 4.

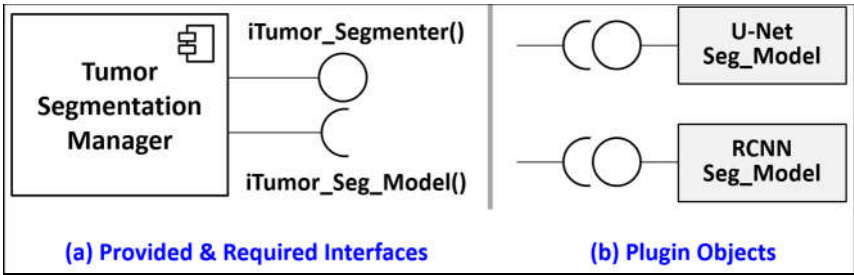


Figure 4. Required Interface and its Plugin Components.

Figure 4a shows two different interfaces of this component: a provided interface, *iTumor_Segmenter()* and a required interface, *iTumor_Seg_Model()* which is implemented in various ways using different CNN segmentation algorithms. Figure 4b shows two plugin objects implemented with different CNN algorithms, which can equally bind to the required interface. This will enable the component *Tumor Segmentation Manager* to work with any valid CNN segmentation models.

4.3. Design of Persistent Datasets

Software systems provide functionality by manipulating relevant data, and some of the data must be stored on permanent data storage or cloud space for persistence. Accordingly, our platform for diagnosing diseases with CECT scans should also manage the set of persistent datasets common among its family of applications."

The persistent data model of the platform has been constructed by applying commonalty and variability analysis on a set of representative family applications including LI-RADS, LUNG-RADS), BI-RADS, and PI-RADS applications. The class diagram capturing persistent datasets of the platform is shown in Figure 5.

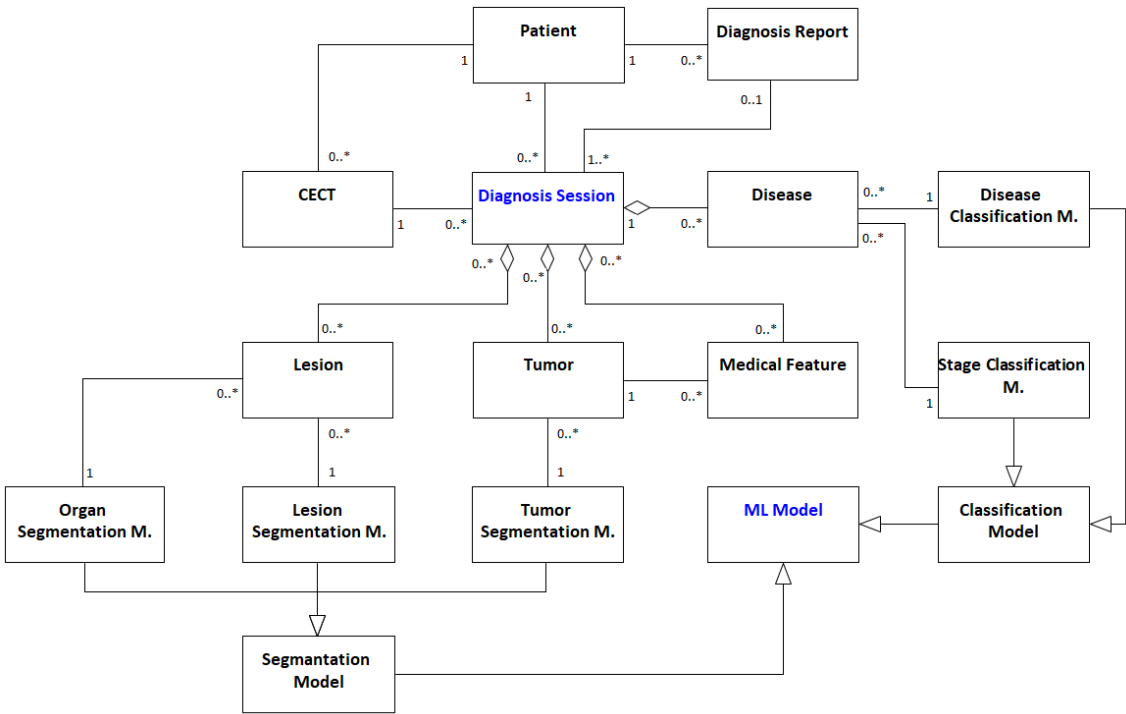


Figure 5. Class Diagram for Persistent Datasets.

The class diagram in the figure organizes persistent objects around two core classes: *Diagnosis Session* and *ML Model*. The *Diagnosis Session* is a logic object class, which captures essential contents

for each disease diagnosis session. Hence, it aggregates other part-classes: *Lesion*, *Tumor*, *Medical Feature*, and *Disease*.

The *ML Model* class captures the meta-information of various machine learning models deployed in the system. This class is specialized into two subclasses: *Segmentation Model* and *Classification Model*. Each of the subclass is further specialized into ML models for specific purposes.

Note that association relationships are defined between classes capturing diagnosis results and the ML models used for diagnosis. For example, the *Tumor* class is associated with the *Tumor Segmentation Model*, making it possible to trace which ML model is utilized for segmenting each specific tumor.

4.4. Design of Unified Diagnosis Process

The process of diagnosing diseases in medical analytics systems using CECT scans reveals a high commonality among different disease types such as Liver disease: Lung disease, Pancreatic disorder, and vascular disease. Hence, this common process should be modeled and designed in one place for promoting effective reuse.

The algorithm design of the unified diagnosis process, *diagnose_disease()*, is shown in Figure 6.

```

1: public diagnose_disease(CT_Scan): diagnosis_report {
2:   // Step 1. Read a patient profile and treatment history.
3:   patient_profile = read_patient_profile();
4:   treatments = read_patient_treatments();
5:
6:   // Step 2. Read CT Scan
7:   ct_scan.read_header();
8:   LIST [CT_Slice] list_slices := null;
9:   list_slices.read(ct_scan);
10:
11:   // Step 3. Segment target organ from CT Slices
12:   organ_seg_model.load();
13:   predictions := organ_seg_model.segment_organ(ct_slices);
14:   slices_organ := generate_organ_slices (predictions);
15:
16:   // Step 4. Segment lesions in target Organ
17:   lesion_seg_model.load();
18:   predictions := lesion_seg_model.segment_lesion(slices_organ);
19:   slices_lesions := generate_lesion_slices (predictions);
20:
21:   // Step 5. Segment tumors from lesions.
22:   tumor_seg_model.load();
23:   predictions := tumor_seg_model.segment_tumor(slices_lesions);
24:   slices_tumors := generate_tumor_slices (predictions);
25:
26:   // Step 6. Analyze medical features of tumors.
27:   LIST [Med_Feature] med_features := null;
28:   med_features_model.load();
29:   predictions := med_features_model.analyze_med_features(slices_tumors);
30:   med_features.add(predictions);
31:
32:   // Step 7. Analyze temporal changes of medical features
33:   LIST [Temporal_Change] temp_change := null;
34:   rnn_model.load();
35:   predictions := rnn_model.analyze_tempo_change(slices_tumors, med_features);
36:   temp_changes.add(predictions);
37:
38:   // Step 8. Classify disease occurrence
39:   LIST [Disease] disease_instances := null;
40:   disease_classifier_model.load();
41:   predictions := disease_classifier_model.classify_disease(slices_tumors,
42:     med_features, temp_changes, treatments);
43:   disease_instances.add(predictions);
44:
45:   // Step 9. Classify disease stages
46:   LIST [Disease_Stage] stage_instances := null;
47:   stage_classifier_model.load();
48:   predictions := stage_classifier_model.classify_stage(slices_tumors,
49:     med_features, temp_changes, treatments);
50:   stage_instances.add(predictions);
51:
52:   // Step 10. Generate diagnosis report
53:   diagnosis_report := generate_diagnosis_report (disease_instances,
54:     stage_instances, slices_tumors);
55:   return (diagnosis_report)
56: }

```

Figure 6. Software Process for Disease Diagnosis with CECT Scans.

The algorithm is defined as a sequence of ten steps. Step 1 is to load a patient’s profile and treatment history. Step 2 is to load a CECT scan of the patient. Steps 3 through 7 are the key tasks of the process, consisting of segmentation, classification, and analytics tasks. Step 8 is to determine occurrences of diseases by referring to the results of steps 3 through 7. Step 10 is to determine the stage of each disease occurrence.

4.5. Design of the Diagnosis Manager

There exists a high commonality of diagnosing various diseases using CECT scans. Modeling this common diagnosis process can lead to more effective and reusable design of the platform. However, there also exists variabilities on some steps in the main process. Both the common and variable features of the diagnosis process should be captured in the design.

We design the diagnosis process by applying the Template Method Pattern, which is represent a single fixed algorithm as the template method and the variability of some steps in the overall algorithm using subclasses and dynamic binding. The design of the Diagnosis Manager by applying this design pattern is shown in Figure 7.

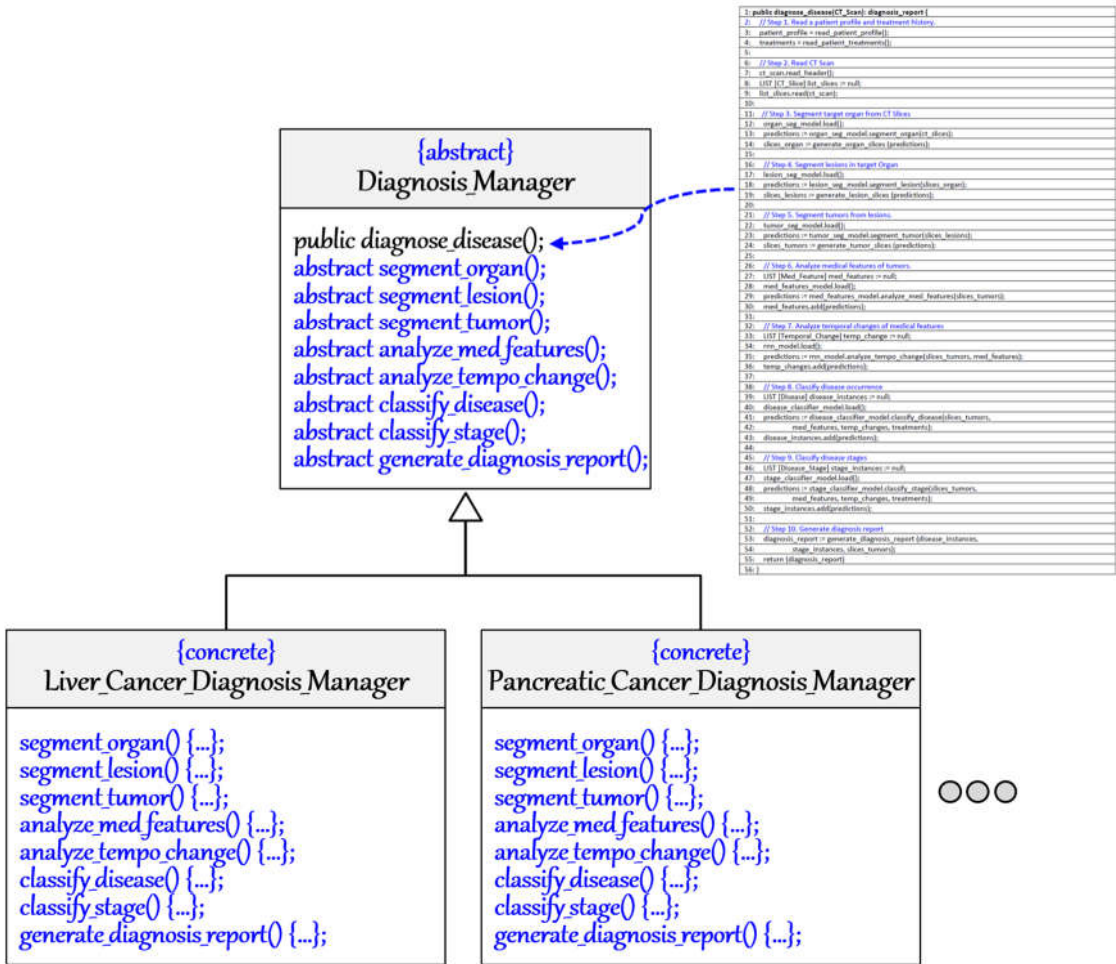


Figure 7. Diagnosis Manager designed with Template Method Pattern.

The class diagram illustrates the application of the Template Method pattern in the design of the *Diagnosis Manager* component. This pattern is characterized by a superclass that defines the skeleton of an algorithm in a template method, consisting of a sequence of method invocations. Some of these method invocations are to abstract methods, referred to as placeholder methods. These placeholder methods, which are designed to be overridden by subclasses, are highlighted in blue in the figure. They represent the customizable parts of the algorithm, allowing for variation in behavior without altering the overall structure or flow of the algorithm.

The figure illustrates three subclasses that implement the abstract methods defined in the superclass. Each subclass adapts these methods to its specific requirements. For instance, the subclass *Liver_Cancer_Diagnosis_Manager* implements the *segment_organ()* method to segment the liver from a given CECT scan. Additionally, it implements the *classify_disease()* method to classify the type of liver cancer, such as *Hepatocellular Carcinoma (HCC)* by applying a standard called *Liver Imaging Reporting and Data System (LI-RADS)* [31]. This classification is based on the analysis of identified tumors, changes over contrast phases, and medical imaging data. The *classify_stage()* method of this subclass will determine a HCC stage using the categories of LR-NC (Not Categorizable), LR-1, LR-2, LR-3, LR-4, LR-5, LR-5V, LR-M, and LR-TR.

Similarly, another subclass *LI_Cancer_Diagnosis_Manager* implements all the abstract methods to its specific requirements for diagnosing *Lung Cancer*. The implementation of the abstract methods in this subclass can be done by applying a standard, *Lung Imaging Reporting and Data System (LUNG-RAD)* [32]. The *classify_stage()* method of this subclass will determine a lung cancer stage using the taxonomy of Category 0 (Incomplete), Category 1 (Negative), Category 2 (Benign Appearance or Behavior), Category 3 (Probably Benign), Category 4A (Suspicious), Category 4B (More Suspicious), and Category 4X (Highest Concern for Malignancy).

This design facilitates the reuse of common diagnostic procedures while allowing for customization to accommodate disease-specific diagnostic methods. The design of *Diagnosis Manager* demonstrates the high reusability of commonality and customizability for variability, which are essential features of any software platforms. This design also provides a high extensibility through a specialization of the superclass for new types of disease diagnosis using CECT.

4.6. Design of Machine Learning Models

The core functionality of this platform is to diagnose diseases by analyzing CECT scans, and the platform relies on several types of ML models in diagnosing diseases. A set of six core ML models has been identified through a comparative analysis of representative standards for analyzing CECT scans.

- Organ Segmentation Model for segmenting a target organ on CECT slices.
- Lesion Segmentation Model for segmenting lesions on CECT slices.
- Tumor Segmentation Model for segmenting tumors on lesions.
- Image Feature Classifier for classifying various image features.
- Disease Type Classifier for classifying diseases from the identified tumors and their medical features.
- Disease Stage Classifier for classifying a stage for the identified diseases.

A challenge in managing ML models for this platform is to handle the high variability of ML algorithms for various diseases and the multiple versions of a ML model trained with different training sets. Therefore, the platform should be designed to support high configurability of training ML models with different algorithms and binding ML models on-the-fly.

Our approach to meet this requirement is to the ML management within the main program using the Bridge Design Pattern. This pattern facilitates system architecture by decoupling the abstraction part of the system from its implementation part. The design of ML management is shown in Figure 8.

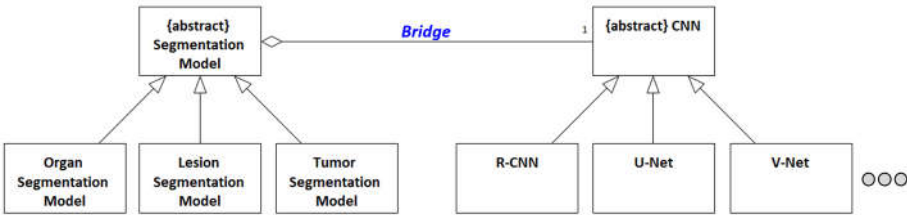


Figure 8. Configuring ML Models by applying Bridge Design Pattern.

The class diagram depicted in the figure shows two abstract classes: *Segmentation Model* and *CNN Segmentation Algorithm*. Each abstract class is specialized into multiple subclasses, which represent valid alternatives at their respective variation points. On the left side of the Bridge, there is a class hierarchy that includes various segmentation models tailored for specific segmentation targets such as Organ, Lesion, and Tumor. Conversely, the right side of the Bridge shows a class hierarchy for a range of CNN-based segmentation algorithms.

This structure effectively demonstrates the decoupling of abstraction (segmentation Models) from implementation (Segmentation Algorithms). This separation allows for the independent evolution of segmentation models and the underlying computational segmentation algorithms.

5. Case Study of Developing Liver Cancer Diagnosis

We present the result of a case study that develops a Liver Cancer Diagnosis system by utilizing the proposed platform.

5.1. HCC Type of Liver Cancer

Liver cancer is considered one of the most life-threatening cancers due to its typically late diagnosis and rapid progression. Hepatocellular Carcinoma (HCC) is the most common type of primary liver cancer, accounting for about 75% to 85% of cases. There exists a widely accepted standard for diagnosing HCC type of liver cancer, called Liver Imaging Reporting and Data System (LI-RADS).

Our case study is to develop a medical image analytics system that particularly diagnose occurrences of HCC using CECT scans. LI-RADS categorizes liver observations into different categories based on their imaging features and the likelihood of representing HCC. The criteria for HCC categorization are summarized in Table 3.

Table 3. Diagnostic Criteria for HCC Liver Cancer.

Category	Probability of HCC	Imaging Features		Management
LR-1	Definitely benign	- Diagnostic features for a benign entity OR - Lesion resolves without treatment		Continue standard surveillance
LR-2	Probably benign	Suggestive, but not diagnostic, features for a benign entity		- Continue standard surveillance - If >1 cm, consider accelerated follow-up, alternative imaging or multidisciplinary discussion
LR-3	Intermediate probability of HCC	- Mass HYPERenhancing in the arterial phase with NO WCT - Mass HYPO- or ISOenhancing in the arterial phase: <20 mm with ≤1 of WCT ≥20 mm with NO WCT		- Variable - At a minimum accelerated follow-up, consider alternative imaging or multidisciplinary discussion
LR-4	Probably HCC	- Mass HYPERenhancing in the arterial phase: <10 mm with ≥1 of WCT 10-19 mm with 1 of WCT ≥20 mm with NO WCT - Mass HYPO- or ISOenhancing in the arterial phase: <20 mm with ≥2 of WCT ≥20 mm with ≥1 of WCT		- Variable - Alternative imaging (if distinct advantage) and/or multidisciplinary discussion
LR-5	Definitely HCC	- Mass HYPERenhancing in the arterial phase: 10-19 mm with ≥2 of WCT ≥20 mm with ≥1 of WCT		Multidisciplinary discussion
LR-5V	Tumor in vein	Definite enhancing soft tissue in vein		- Multidisciplinary discussion - Contraindication to liver transplant
LR-M	Probable malignancy, but non-specific for HCC	<i>Favor other malignancy</i> - Arterial rim HYPERenhancement - Central delayed phase enhancement - Concentric enhancement - Peripheral washout - Hepatic retraction - Biliary obstruction greater than expected for size - Targetoid on DWI or hepatobiliary phase	<i>Favor HCC</i> - Diffuse arterial HYPERenhancement - Delayed washout - Capsule - Distinct rim - Intralesional fat - Diffuse T1 HYPERintensity - Tumor in vein - Nodule-in-nodule - Mosaic architecture - Spontaneous hemorrhage	Variable, may include follow-up, alternative imaging, biopsy, treatment, and/or multidisciplinary discussion

The imaging features listed in the third column of the table can be directly determined based on the results of organ, lesion, and tumor segmentations as well as medical features classified by the platform.

5.2. Customizing the Platform for HCC Diagnosis

We adopted the proposed platform as the foundation and specialized its design for HCC Diagnosis system as follows.

5.2.1. Designing the Schematic Architecture

The schematic architecture of the platform as depicted in Figure 2 was devised to be generic and common among disease diagnosis systems using CECT. Hence, the architecture was applied to HCC Diagnosis system without any modifications.

The allocation of the 12 functional components on Core Control layer and Plugin Control layer was applicable to this target system as well. No additional components were needed in developing the HCC diagnosis system.

5.2.2. Applying the Diagnosis Process for HCC Diagnosis

The platform defines a diagnosis process as a sequence of ten steps, as shown in Figure 6. The control flow of this diagnosis process is also applicable to diagnosing HCC according to the LI-RADS standard. Hence, no change was made on the diagnosis process.

5.3.3. Refining the Diagnosis Manager for HCC Diagnosis

The platform defines the design of *Diagnosis Manager* by applying the Template Method pattern, as shown in Figure 7. To specialize the 8 abstract methods defined in the superclass *Diagnosis_Manager* were implemented in a subclass for the purpose of diagnosing HCC.

The design of *HCC Diagnosis Manager* by specializing *Diagnosis Manager* is shown in Figure 9.

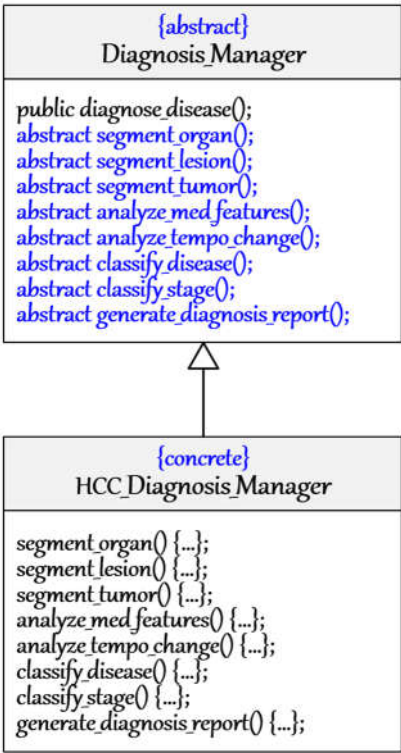


Figure 9. Specializing Diagnosis Manager for HCC Liver Cancer Diagnosis.

The class *HCC_Diagnosis_Manager* is defined as a subclass, and it implements the 8 abstracts for HCC diagnosis by applying the guidelines of LI-RADS standard. We use numpy library for handling arrays and keras for leveraging deep learning models.

For example, the implementation of the method for segmenting tumors from lesions found in CT slices, *segment_tumor()* is shown in Figure 10.

```

1. import numpy as np
2. import cv2
3. from tensorflow.keras.models import load_model
4.
5. # Load the pre-trained CNN model for tumor segmentation
6. tumor_seg_model = load_model(file_slices_lesions)
7.
8. # Preprocess the CT slice to be suitable for the model input.
9. def preprocess_image(ct_slice):
10.     # Resize the image to the required size for the model, e.g., 256 × 256
11.     ct_slice_resized = cv2.resize(ct_slice, (256, 256))
12.
13.     # Normalize the image to the range the model expects, e.g., 0-1
14.     ct_slice_normalized = ct_slice_resized/255.0
15.
16.     # Expand the dimensions to fit the model input shape, e.g., (1, 256, 256, 1)
17.     ct_slice_expanded = np.expand_dims(ct_slice_normalized, axis=0)
18. return ct_slice_expanded
19.
20. # Segment tumor in a CT slice using the CNN-based model.
21. def segment_tumor(ct_slice):
22.     # Preprocess the CT slice
23.     processed_slice = preprocess_image(ct_slice)
24.
25.     # Perform prediction (segmentation)
26.     segmentation = tumor_seg_model.predict(processed_slice)
27.
28.     # Post-process to create a binary mask, in the range [0, 1]
29.     segmentation_mask = (segmentation [0] > 0.5).astype(np.uint8)
30. return segmentation_mask

```

Figure 10. Python Program for Specializing *segment_tumor()*;

This program uses the same reference for a tumor segmentation model, *tumor_seg_model*. It performs three tasks: reading the list of pre-segmented lesions in line #6, preprocessing CT slices for segmentation in line #9, and segmenting tumors by invoking '*predict()*' method in line #26.

5.2.4. Applying the Database for HCC Diagnosis System

The class diagram capturing persistent datasets in the platform is designed to be generic and common among the family of diagnosis applications. This design is also applicable to the data persistence of HCC Diagnosis system, requiring no change on the persistence design.

5.3. Training ML Models for HCC Diagnosis

5.3.1. Trainset of Labeled CECT Scans

We acquired the training set, i.e., a set of labeled CECT scans, from two sources: Loma Linda University Medical Center [35] and Radiopaedia. The CECT scans from the Medical Center were manually labeled by an attending physician and staff members in Vascular and Interventional Radiology. The labels include image masks of lesions and tumors on the liver regions on CT slices and all the necessary features specified in LI-RADS.

5.3.2. Training ML Models for HCC

The six essential ML models of the platform were applicable to HCC diagnosis. We developed an additional ML model for analyzing medical features presented in tumors. The types of ML models implemented in this case study are summarized in Table 4.

Table 4. Machine Learning Models for HCC Diagnosis.

ML Models	Organ Segmentation Model	Lesion Segmentation Model	Tumor Segmentation Model	Hepatic Segmentation Model	Image Feature Classifier	HCC Type Classifier	HCC Stage Classifier
ML Category	Segmentation				Classification		
ML Algorithm	U-Net				CNN	LSTM	SVM
Functional Components	Organ Segmentation Manager	Lesion Segmentation Manager	Tumor Segmentation Manager	Lesion/Tumor Segmentation Manager	Medical Feature Analyzer	Disease Type Classifier	Disease Stage Classifier
Prediction	Liver in CT Slices	Lesions in Liver	Tumors in Lesions	Hepatic Segments	List of Image Features	HCC Occurrence	Stage of HCC

The selection of ML algorithms for segmentation and classification has been made through extensive experiments of training the models with other candidate models.

5.3.3. Integrating Image Feature Classifier and HCC Type Classifier

The appearance of HCC on CT changes over time as the tumor grows and develops a blood supply. These changes are particularly evident when contrast agents are used like CECT. The contrast enhancement pattern over different phases (arterial, venous, delayed) provides important diagnostic information. HCC typically shows hyperenhancement in the arterial phase (due to its increased arterial blood supply) and washout in the venous or delayed phases (due to its lack of portal venous blood supply).

We applied the Long Short-Term Memory (LSTM) algorithm, a type of RNN, to capture this temporal changes. The input to the LSTM consists of a sequence of medical features sets observed on different phases of the same detected tumor, where the medical features are identified by CNN Image Feature Classifier. Hence, we integrated the CNN image feature classifier (Step 6 in the process) and LSTM HCC Type classifier (Step 7 in the process).

The structure of the integration and information flow in this integration are shown in Figure 11.

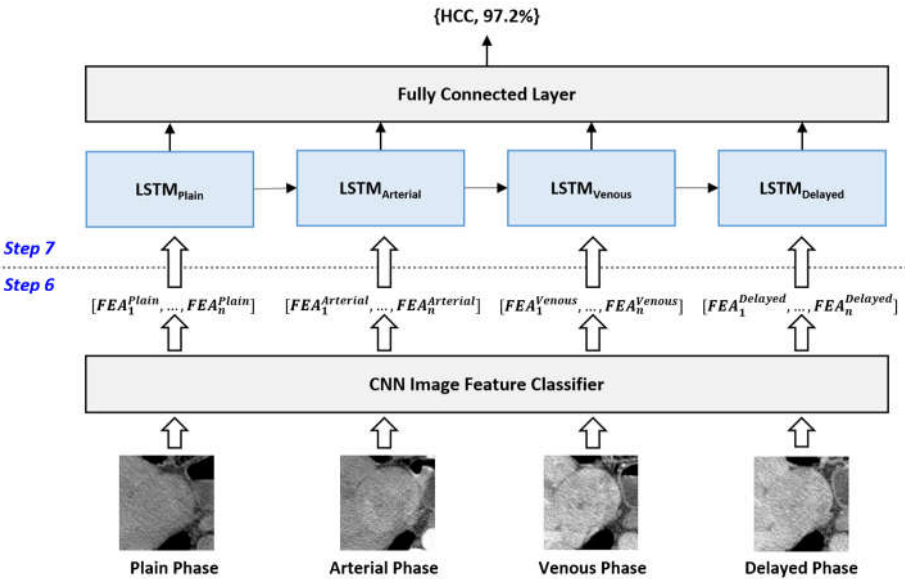


Figure 11. Integrating CNN Classifier to LSTM Classifier for HCC.

5.4. Implementing HCC Diagnosis System

We implemented the HCC Diagnosis System in Python 3.11, and trained the ML models using Scikit-learn and TensorFlow 2.4 with the Keras API.

5.4.1. User Interface of the System

The user interface of the system was developed as a web-based interface, and a screen of the interface is shown in Figure 12.

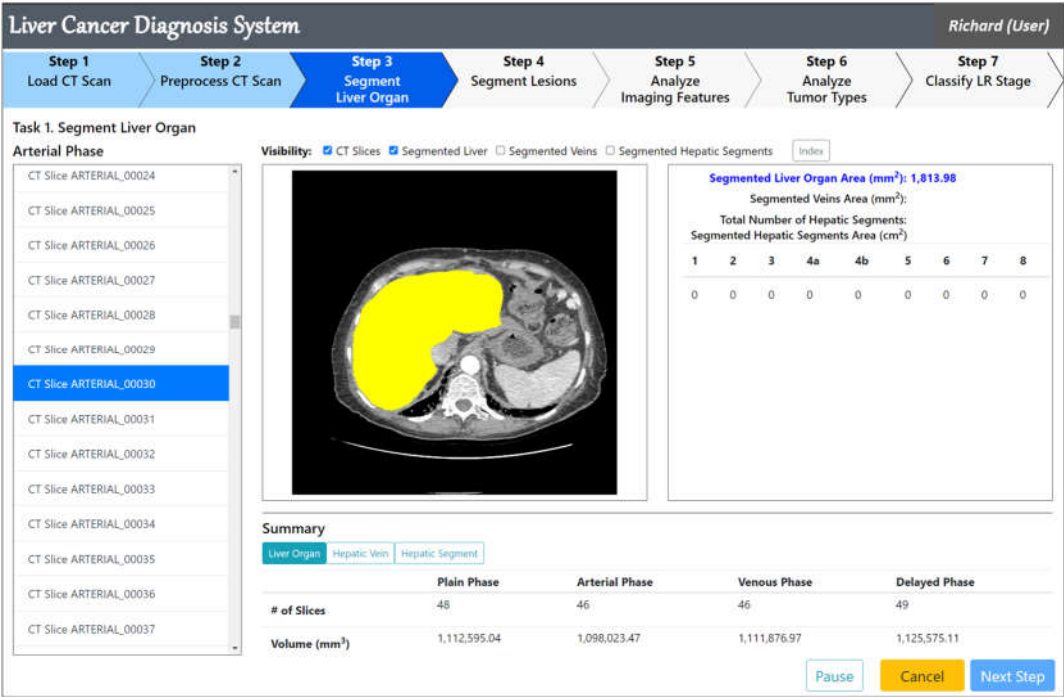


Figure 12. Web User Interface of HCC Diagnosis System.

The system displays the seven steps of HCC Diagnosis at the top of the screen by adhering to the LI-RADS standard. The list of CT slices in a CT scan is displayed on the left side of the screen, with segmented organ highlighted in yellow. The summary of organ segmentation is shown at the bottom of the screen. The volume of organ is also computed and displaced in the table.

We designed the user interface to be intuitive to self-explanatory. Additionally, the user interface was designed to be comprehensive, displaying all intermediate analytics results.

5.4.2. Experiments with CECT Test Set

Software testing is an effective way for validating the functionality and features of software platforms. We created a test set using the CECT scans from Loma Linda University Medical Center using the typical distribution of datasets into 80% of training set and 20 of test set.

The result of running HCC diagnosis using this implementation is shown in Figure 13.

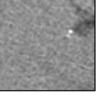
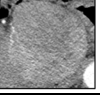
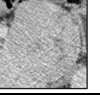
	Plain	Arterial	Venous	Delayed	Major Features	Size (mm)	Location	Stage
(a)					Nonrim APHE	9.12	Segment II	LR-3 (93.1%)
(b)					Nonrim APHE, Washout	8.03	Segment III	LR-4 (96.2%)
(c)					Nonrim APHE, Washout	9.38	Segment VI	LR-4 (94.6%)
(d)					Nonrim APHE, Washout	22.15	Segment IVa	LR-5 (98.4%)
(e)					Nonrim APHE, Washout	80.53	Segment I	LR-5 (97.8%)

Figure 13. Example of HCC Diagnosis Report.

The system output shows occurrences of five HCC, where each HCC is shown the temporal changes of the tumor, major medical features, the size of tumor, its location as Hepatic Segment ID, and determined stage.

5.4.3. Measuring Performance of Machine Learning Models

The performance of ML models itself is not much relevant to the quality of platform design since the models are trained outside of the platform by using the disease-specific datasets. However, we measured the performance of 7 ML models for HCC diagnosis as the system implementation is available.

The average performance of the 7 ML models is shown in Figure 14.

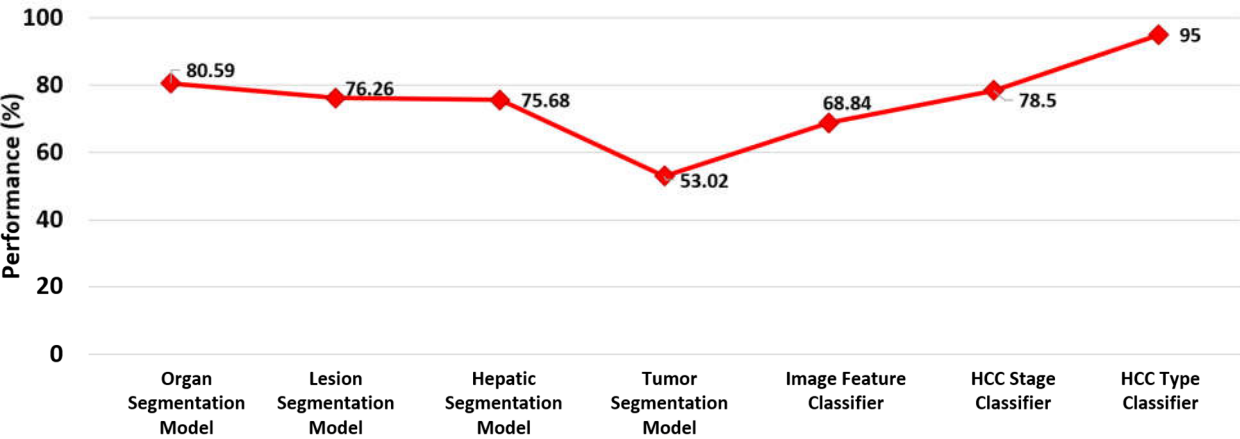


Figure 14. Performance of ML Models for HCC Diagnosis.

The performance measures for the ML models vary due to the intrinsic characteristics of each prediction task. The first three segmentation models show higher performance than the Tumor Segmentation Model. This difference can be attributed to three factors: the size of the target objects, the vividness of the target objects as they appear in CT slices, and the intrinsic nature of visual ambiguity and blurriness of tumors.

6. Assessment of the Platform

Evaluating the development of software platforms is a multi-faceted process that involves assessing various qualitative and quantitative aspects of the software.

6.1. Evaluating Functional Coverage of the Platform

The *Functional Coverage* of software platforms refers to the extent to which the platform's features and capabilities meet the various requirements of potential applications that can be built with the platform [36]. It ensures that all necessary functionalities are implemented and can handle the diverse needs of medical diagnostics using CECT scans.

We define a metric *Functional Coverage (FC)* for the Functional Coverage of the platform as the following.

$$FC = \frac{N_f}{N_t}$$

N_f is the number of features implemented by the platform and N_t is the total number of features required by the family of medical diagnostic applications using CECT scans. The value range of CR is from 0 to 1, where 0 indicates that none of the required features have been implemented and 1 indicates that all required features have been fully implemented.

The value of N_f is computed as the sum of the four coverage-relevant categories:

- The number of functional components that fulfill the functional requirements.
- The number of persistent object classes that manage the persistent datasets.
- The number of diagnostic tasks specified in the main diagnosis process.
- The number of machine learning models required by the functional components.

The functional components of the platform were derived directly from the use case diagram of the platform. Hence, the set of 12 functional components effectively fulfills the requirements.

The persistent object classes were derived from the data manipulated by each functional component. Hence, the set of 16 persistent object classes represents the persistent datasets managed by medical image analytics systems.

The 8 steps, 3 through 10, of the defined diagnostic process are directly mapped to the diagnosis tasks for diagnosing the target diseases with the CECT, as defined in various standards such as LI-RADS [31] and LUNG-RADS [32].

The machine learning models needed for diagnosing diseases were derived by considering the intrinsic nature of the required diagnostic tasks, as summarized in Table 2. A total of 6 machine learning models, trained with CNN, RNN, and SVM classification algorithms, sufficiently assist the required diagnostic tasks.

Accordingly, the N_f for the platform is computed to 42, as shown in Table 5.

Table 5. Computing Functional Coverage (FC) Index.

Categories of Features	# of Features (Platform)	# of Features (Liver Cancer System)	Functional Coverage (FC)
Functional Components	12	13	92.3%
Persistent Object Classes	16	16	100.0%
Tasks in Diagnosis Process	8	9	88.9%
ML Models used in Diagnosis	6	7	85.7%
Sum	42	45	93.3%
	N_f	N_t	

All 42 features of the platform have been applicable to developing the Liver Cancer Diagnosis System with the platform. In addition, the Liver Cancer Diagnosis System required additional features to fulfill the diagnosis process and guidelines of the LI-RADS [32].

- Additional Functional Component: *Liver Lesion Tracker*

This is a specialized software component designed to monitor and document changes in liver lesions over time. This functionality is crucial for the effective management and treatment of liver diseases, particularly hepatocellular carcinoma (HCC), as specified in the LI-RADS framework.

- Additional Task in Diagnosis Process: *Specializing Step 9 for Tracking Lesions*

Step 9 of the diagnosis process in section 4.4 has been specializing to support the additional functionality of tracking lesions on Liver.

- Additional ML Model in Diagnosis Process: *Hepatic Segmentation Model*

This model is required for segmenting hepatic segments on liver organ. These models are trained using U-Net algorithm, to utilize the encoder-decoder feature that allows for precise localization and segmentation.

The numbers of these additional features required for Liver Cancer Diagnosis system are shown in the third column of Table 5. As shown in the table, the value of N_i , the number of features required for this target system, becomes 45. This will yield 93.3% of the *Functional Coverage (FC)* for the case study of constructing the LI-RADS application.

6.2. Evaluating Feature Satisfaction Index

The *Feature Satisfaction Index (FSI)* is a quantitative metric used to evaluate the degree to which a software platform's features meet the set of common functional and non-functional requirements of potential applications that can be built with the platform [37]. This metric takes into account both the importance (weight) of each feature and how well each feature has been implemented (compliance score), providing a comprehensive assessment of overall feature satisfaction.

We define a metric *Feature Satisfaction (FS)* for the Feature Satisfaction of the platform as the following.

$$FS = \frac{\sum_{i=1}^{N_f} w_i \cdot c_i}{\sum_{i=1}^{N_f} w_i}$$

where the terms are defined as the following.

- N_f is the total number of features provided by the platform
- w_i is the weight assigned to each *feature i*, representing the relative importance of each feature in the context of the platform's overall objectives. Its value ranges between 0 and 1. Features critical to the platform's functionality are assigned higher weights, while less critical features receive lower weights.
- c_i is the compliance score of each *feature i*, indicating how well the feature is implemented. The compliance score assesses how well each feature meets its requirements. Its value ranges between 0 and 1. The value 0 indicates that the feature does not meet the requirements at all, and the value 1 indicates full compliance.

Through the case study of developing Liver Cancer Diagnosis System using the LI-RADS standard, we measured the compliance scores for the 42 features of the platform and synthesized their values into the metric, *FS*. This result is summarized in Table 6.

Table 6. Computing Feature Satisfaction (FS) Index.

Categories of Features	List of Features	Weight (W _i)	Compliance Score (C _i)	W _i * C _i	Feature Satisfaction (FS)
Functional Components	Patient Profile Manager	0.01	1	0.01	100%
	Symptom Manager	0.01	1	0.01	100%
	Medical Image Manager	0.03	1	0.03	100%
	Feedback Manager	0.01	1	0.01	100%
	Diagnosis Session Manager	0.015	0.8	0.012	80%
	Diagnosis Report Generator	0.015	0.9	0.0135	90%
	Organ Segmentation Manager	0.03	1	0.03	100%
	Lesion Segmentation Manager	0.03	1	0.03	100%
	Tumor Segmentation Manager	0.03	1	0.03	100%
	Medical Feature Analyzer	0.03	0.7	0.021	70%
	Disease Type Classifier	0.02	0.9	0.018	90%
	Disease Stage Classifier	0.02	0.9	0.018	90%
	Sub-total	0.25		0.2325	93.0%
Persistent Object Classes	Patient	0.01	1	0.01	100%
	Diagnosis Report	0.015	1	0.015	100%
	CECT	0.01	1	0.01	100%
	Diagnosis Session	0.025	0.8	0.02	80%
	Disease	0.025	1	0.025	100%
	Lesion	0.025	1	0.025	100%
	Tumor	0.025	1	0.025	100%
	Medical Feature	0.025	0.7	0.0175	70%
	Organ Segmentation Model	0.01	1	0.01	100%
	Lesion Segmentation Model	0.01	1	0.01	100%
	Tumor Segmentation Model	0.01	1	0.01	100%
	Disease Classification Model	0.01	1	0.01	100%
	Stage Classification Model	0.01	1	0.01	100%
	Segmentation Model	0.01	1	0.01	100%
	Machine Learning (ML) Model	0.02	0.9	0.018	90%
	Classification Model	0.01	1	0.01	100%
	Sub-total	0.25		0.2355	94.2%
Tasks in Diagnosis Process	Step 3. Segment target organ	0.035	1	0.035	100%
	Step 4. Segment lesions	0.035	1	0.035	100%
	Step 5. Segment tumors	0.035	0.8	0.028	80%
	Step 6. Analyze medical features	0.035	0.7	0.0245	70%
	Step 7. Analyze temporal changes	0.035	0.8	0.028	80%
	Step 8. Classify disease occurrence	0.03	0.8	0.024	80%
	Step 9. Classify disease stages	0.03	0.9	0.027	90%
	Step 10. Generate diagnosis report	0.015	1	0.015	100%
	Sub-total	0.25		0.2165	86.6%
ML Models used in Diagnosis	Organ Segmentation Model	0.04	1	0.04	100%
	Lesion Segmentation Model	0.05	1	0.05	100%
	Tumor Segmentation Model	0.05	1	0.05	100%
	Image Feature Classifier	0.05	0.7	0.035	70%
	Disease Type Classifier	0.03	0.9	0.027	90%
	Disease Stage Classifier	0.03	0.9	0.027	90%
	Sub-total	0.25		0.229	91.6%
	GRAND TOTAL	1		0.9135	91.4%
					FS

Each category of the features is given an equal overall weight of 0.25, i.e., 25%. The weights of the 42 features are defined by the authors and developers who implement the platform. The weights were determined by considering the relevance and impact of each feature on (1) the diagnostic accuracy, (2) clinical significance, and (3) their contribution to the overall performance of the ML models. These considerations include expert domain knowledge, statistical analyses of feature importance, and empirical results from extensive testing and validation using diverse datasets.

The compliance scores of the 42 features were determined mainly by the developers who implemented the case study system, Liver Cancer Diagnosis System. The determination was based on a combination of factors including (1) clinical relevance, (2) statistical significance, (3) empirical testing, (4) expert consensus, and (5) technical feasibility. This multifaceted approach ensures that the compliance scores reflect a comprehensive and balanced assessment of each feature's value in diagnosing liver cancer accurately and effectively.

As shown in the table, the overall value of Feature Satisfaction (FS) was computed as 91.4%, which we believe is considerably high for a software platform in the medical application domain. We noticed that the FS value for the feature category of Diagnosis Process measured at 86.6%, the lowest among the four categories. This was mainly due to the high variability in the diagnosis processes for (1) different diseases and (2) their relevant body organs and (3) standard guidelines governing the diagnosis processes.

7. Concluding Remarks

In this paper, we have introduced a comprehensive machine learning-based software platform designed to address the complexities and challenges associated with diagnosing diseases using CECT scans. The platform provides a unified software process that can fully automate the diagnosis practices of various diseases using CECT scans. In addition, the platform effectively integrates multiple machine learning models, including Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Support Vector Machines (SVMs), to perform tasks such as organ, lesion, and tumor segmentation, as well as disease classification and staging.

As the key software strategy, we leveraged the Open-Closed Principle (OCP) to develop a highly configurable and extendable system architecture, ensuring that common features remain stable while allowing for customization to meet the specific needs of various diagnostic applications. The platform's architecture, based on the Microkernel style, separates core control functionalities from plugin control functionalities, enhancing modularity and scalability.

Through a detailed case study, we demonstrated the platform's applicability by developing a Liver Cancer Diagnosis System that adheres to the LI-RADS standard. The case study highlighted the platform's ability to support the end-to-end diagnostic process, from CT scan acquisition to generating comprehensive diagnosis reports.

We evaluated the proposed platform using two evaluation metrics: Functional Coverage (FC) and Feature Satisfaction (FS). Each of these evaluation metrics was applied to the case study of developing a Liver Cancer Diagnosis System. Our platform achieved an overall FC score of 93.3% and an overall FS score of 91.4%, indicating its robustness and effectiveness in the medical application domain.

The key contributions of our research can be summarized in the following three folds.

- *Unified Software Process of Diagnosis:* Establishing a unified software process for CECT-based disease diagnosis that is applicable across multiple disease types.
- *Integration of Essential ML Models:* Identifying and integrating six core machine learning models to support various diagnostic tasks, thereby reducing the technical complexity of developing high-performing diagnostic systems.
- *Configurable and Extensible Design:* Designing the platform to be highly configurable and extendable, enabling efficient customization for different diagnostic requirements.

Our future work should focus on expanding the types of diseases that can be diagnosed using our platform, integrating more dynamic and self-learning algorithms, and enhancing the platform's ability to manage and analyze large datasets from diverse populations. We note that conducting a case study of developing an application with the proposed platform requires a tremendous amount of time and effort. This is primarily due to the required efforts for acquiring a sufficient number of CECT scans, labelling the CT slices for training purposes, training the required ML models, and fine-tuning the model for acceptable performances. By cooperating with university hospitals and private clinics, we aim to conduct each case study within approximately six months.

Author Contributions: Jennifer Jin and Mira Kim conceptualized the research scope and coordinated the co-authoring of the paper and peer review. Soo Dong Kim designed the software platform and lead the implementation of the platform and the case study system. Daniel Jin was responsible for governing software process and methods that automate the application of LI-RADS. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Tao, L.L.; Dai, Y.; Yin, W.; Chen, J. A case report of a renal anastomosing hemangioma and a literature review: an unusual variant histologically mimicking angiosarcoma. *Diagn Pathol.* **2014**, *9*, 159.
2. Wang, B.; Jin, S.; Yan, Q.; Xu, H.; Luo, C.; Wei, L.; Zhao, W.; Hou, X.; Ma, W.; Xu, Z.; Zheng, Z.; Sun, W.; Lan, L.; Zhang, W.; Mu, X.; Shi, C.; Wang, Z.; Lee, J.; Jin, Z.; Lin, M.; Jin, H.; Zhang, L.; Guo, J.; Zhao, B.; Ren, Z.; Wang, S.; Xu, W.; Wang, X.; Wang, J.; You, Z.; Dong, J. AI-assisted CT imaging analysis for COVID-19 screening: Building and deploying a medical AI system. *Appl Soft Comput.* **2021**, *98*, 106897.
3. Kouanou, A.T.; Tchiotso, D.; Kengne, R.; Zephirin, D.T.; Armele, N.M.A.; Tchinda, R. An optimal big data workflow for biomedical image analysis, *Informatics in Medicine Unlocked.* **2018**, *11*, 68-74.
4. Mansour, R.F.; Escorcia-Gutierrez, J.; Gamarra, M.; Díaz, V.G.; Gupta, D. Kumar, S.; Artificial intelligence with big data analytics-based brain intracranial hemorrhage e-diagnosis using CT images. *Neural Comput & Applic.* **2023**, *35*, 16037–16049.
5. Dhar, T.; Dey, N.; Borra, S.; Sherratt, R.S.; Challenges of Deep Learning in Medical Image Analysis—Improving Explainability and Trust. *IEEE Transactions on Technology and Society.* **2023**, *4*, 1, 68-75.
6. Xie, Y.; Chen, M.; Kao, D.; Gao, G.; Chen, X.A. CheXplain: Enabling Physicians to Explore and Understand Data-Driven, AI-Enabled Medical Imaging Analysis. In Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20). **2020**, Association for Computing Machinery, New York, NY, USA, 1–13.
7. Olveres, J.; González, G.; Torres, F.; Moreno-Tagle, J.C.; Carbajal-Degante, E.; Valencia-Rodríguez, A.; Méndez-Sánchez, N.; Escalante-Ramírez, B. What is new in computer vision and artificial intelligence in medical image analysis applications. *Quant Imaging Med Surg.* **2021**, *11*, 8, 3830-3853.
8. Lei, Y.; Yin, M.; Yu, M.; Yu, J.; Zeng, S.; Lv, W.; Li, J.; Ye, H.; Cui, X.; Dietrich, C.F. Artificial Intelligence in Medical Imaging of the Breast. *Frontiers in Oncology.* **2021**, *11*.
9. Lutnick, B.; Ginley, B.; Govind, D.; McGarry, S.D.; LaViolette, P.S.; Yacoub, R.; Jain, S.; Tomaszewski, J.D.; Jen, K.; Sader, P. An integrated iterative annotation technique for easing neural network training in medical image analysis. *Nat Mach Intell.* **2019**, *1*, 112–119.
10. Mehta, N.; Pandit, A.; Shukla, S. Transforming healthcare with big data analytics and artificial intelligence: A systematic mapping study. *Jour of Bio Inf.* **2019**, *100*, 103311.
11. Ker, J.; Wang, L.; Rao, J.; Lim, T. Deep Learning Applications in Medical Image Analysis. *IEEE Access.* **2018**, *6*, 9375-9389.
12. Chen, M.; Shi, X.; Zhang, Y.; Wu, D.; Guizani, M. Deep Feature Learning for Medical Image Analysis with Convolutional Autoencoder Neural Network. *IEEE Transactions on Big Data.* **2021**, *7*, 4, 750-758.
13. Willemink, M. J.; Koszek, W. A.; Hardell, C.; Wu, J.; Fleischmann, D.; Harvey, H.; Folio, L.R.; Summers, R.M.; Rubin, D. L.; Lungren, M. P. Preparing Medical Imaging Data for Machine Learning. *Radiology.* **2020**, *295*, 1, 4-15.
14. Shi, F.; Wang, J.; Shi, J.; Ziyang, W.; Wang, Q.; Tang, Z.; He, K.; Shi, Y.; Shen, D. Review of Artificial Intelligence Techniques in Imaging Data Acquisition, Segmentation, and Diagnosis for COVID-19. *IEEE Reviews in Biomedical Engineering.* **2021**, *14*, 4-15.
15. Gosalia, A.; Haramati, L.B.; Sheth, M.P.; Spindola-Franco, H. CT detection of acute myocardial infarction. *AJR Am J Roentgenol.* **2004**, *182*, 6, 1563-6.
16. Gono, W.; Hayashi, T.Y.; Okuma, H.; Akahane, M.; Nakai, Y.; Mizuno, S.; Tateishi, R.; Isayama, H.; Koike, K.; Ohtomo, K. Development of pancreatic cancer is predictable well in advance using contrast-enhanced CT: a case-cohort study. *Eur Radiol.* **2017**, *27*, 12, 4941-4950.
17. Chang, E.H.; Chong, W.K.; Kasoji, S.K.; Fielding, J.R.; Altun, E.; Mullin, L.B.; Kim, J.I.; Fine, J.P.; Dayton, P.A.; Rathmell, W.K. Diagnostic accuracy of contrast-enhanced ultrasound for characterization of kidney lesions in patients with and without chronic kidney disease. *BMC Nephrol.* **2017**, *18*, 1, 266.
18. Cayet, S.; Pasco, J.; Dujardin, F.; Besson, M.; Orain, I.; De Muret, A.; Miquelstorena-Standley, E.; Thierry, J.; Genet, T.; Le Bayon, A.G. Diagnostic performance of contrast-enhanced CT-scan in sinusoidal obstruction syndrome induced by chemotherapy of colorectal liver metastases: Radio-pathological correlation. *Eur J Radiol.* **2017**, *94*, 180-190.
19. Rawson, J.V.; Pelletier, A.L. When to order contrast-enhanced CT. *American family physician.* **2013**, *88*, 5, 312-316.

20. Ohno, Y.; Koyama, H.; Lee, H.Y.; Miura, S.; Yoshikawa, T.; Sugimura, K. Contrast-enhanced CT- and MRI-based perfusion assessment for pulmonary diseases: basics and clinical applications. *Diagn Interv Radiol*. **2016**, 22, 5, 407-421.
21. Scheffel, H.; Stolzmann, P.; Frauenfelder, T.; Schertler, T.; Desbiolles, L.; Leschka, S.; Marincek, B.; Alkadhi, H.. Dual-energy contrast-enhanced computed tomography for the detection of urinary stone disease. *Investigative radiology*. **2007**, 42, 12, 823-829.
22. Fletcher, J.G.; Johnson, C.D.; Krueger, W.R.; Ahlquist, D.A.; Nelson, H.; Ilstrup, D.; Harmsen, W.S.; Corcoran, K.E. Contrast-enhanced CT colonography in recurrent colorectal carcinoma: feasibility of simultaneous evaluation for metastatic disease, local recurrence, and metachronous neoplasia in colorectal carcinoma. *American Journal of Roentgenology*. **2002**, 178, 2, 283-290.
23. Iavarone, M.; Piscaglia, F.; Vavassori, S.; Galassi, M.; Sangiovanni, A.; Venerandi, L.; Forzenigo, L.V.; Golfieri, R.; Bolondi, L.; Colombo, M. Contrast enhanced CT-scan to diagnose intrahepatic cholangiocarcinoma in patients with cirrhosis. *Journ of Hepato*. **2013**, 58, 6, 1188-1193.
24. Chen, L.D.; Xu, H.X.; Xie, X.Y.; Lu, M.D.; Xu, Z.F.; Liu, G.J.; Liang, J.Y.; Lin, M.X. Enhancement patterns of intrahepatic cholangiocarcinoma: comparison between contrast-enhanced ultrasound and contrast-enhanced CT. *The British journal of radiology*. 2008, 81, 971, 881-889.
25. Prionas, N.D., Lindfors, K.K., Ray, S., Huang, S.Y., Beckett, L.A., Monsky, W.L. and Boone, J.M., 2010. Contrast-enhanced dedicated breast CT: initial clinical experience. *Radiology*, 256(3), pp.714-723.
26. Cha, S.; Abusharekh, A.; Abidi, S.S. Towards a 'Big' Health Data Analytics Platform. *IEEE First International Conference on Big Data Computing Service and Applications*, **2015**, 233-24.
27. Vuppala, S.K.; Dinesh, M.S.; Viswanathan, S.; Ramachandran, G.; Bussa, N.; Geetha, M. Cloud based big data platform for image analytics. In *2017 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM)*. **2017**, 11-18.
28. Chrimes, D.; Moa, B.; Zamani, H.; Kuo, M.H. Interactive healthcare big data analytics platform under simulated performance. In *2016 IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech)*. **2016**, pp. 811-818.
29. Allen, W.J., Gabr, R.E., Tefera, G.B., Pednekar, A.S., Vaughn, M.W. and Narayana, P.A., 2017. Platform for automated real-time high performance analytics on medical image data. *IEEE journal of biomedical and health informatics*, 22(2), pp.318-324.
30. Kaur, J.; Mann, K.S. AI based healthcare platform for real time, predictive and prescriptive analytics using reactive programming. In *Journal of Physics: Conference Series*. 2017, 933, 1, 012010.
31. American College of Radiology, Liver Imaging Reporting and Data System (LI-RADS). 2018.
32. American College of Radiology, Lung Imaging Reporting and Data System, (LUNG-RAD), 2014.
33. Meyer, Bertrand. Object-Oriented Software Construction. Prentice Hall, 1988.
34. S. Kim, J. Her, and S. Chang, "A theoretical foundation of variability in component-based development," *Information and Software Technology*, Vol.47, No. 10, July 2005.
35. Loma Linda University Medical Center. Department of Radiology, Loma Linda, CA, USA.
36. R. S, Pressman, *Software Engineering: A Practitioner's Approach*. McGraw-Hill, 2014.
37. S. H. Kan, *Metrics and Models in Software Quality Engineering*. Addison-Wesley, 2003.
38. American College of Radiology, Breast Imaging Reporting and Data System (BI-RADS), 2013.
39. American College of Radiology, Prostate Imaging Reporting and Data System (PI-RADS), 2019.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.