MHub.ai：用于医学影像人工智能模型的简易、标准化且可复现的平台

Leonard Nürnberg[1,2,3], Dennis Bontempi[1,2,3], Suraj Pai[1,2,3], Curtis Lisle[4], Steve Pieper[5], Ron Kikinis[6], Sil van de Leemput[7], Rahul Soni[8], Gowtham Murugesan[8], Cosmin Ciausu[1,2,3], Miriam Groeneveld[7], Felix J. Dorfner[9,10], Jue Jiang[11], Aneesh Rangnekar[11,], Harini Veeraraghavan[11], Joeran S. Bosma[7], Keno Bressem[12,13], Raymond Mak[1,3], Andrey Fedorov[6], Hugo JWL Aerts[1,2,3]

[1] 医学人工智能（AIM）项目，Mass General Brigham，哈佛医学院，美国马萨诸塞州波士顿。[2] 马斯特里赫特大学放射学与核医学系，CARIM 与 GROW，荷兰马斯特里赫特。[3] 放射肿瘤科，达纳-法伯癌症研究所，布莱根妇女医院，哈佛医学院

,

美国马萨诸塞州波士顿。[4] 美国佛罗里达州梅特兰，KnowledgeVis, LLC。[5] 美国马萨诸塞州剑桥，Isomics, Inc.[6] 美国马萨诸塞州波士顿，布莱根妇女医院、哈佛医学院放射科。[7] 荷兰奈梅亨，拉德堡德大学医学中心影像科[8] 美国密歇根州大急流城，BAMF Health LLC[9] 德国柏林，柏林夏里特医学院放射科（该院为柏林自由大学和柏林洪堡大学的法人成员）[10]阿西努拉·A·马蒂诺斯生物医学影像中心，马萨诸塞州总医院和哈佛医学院，美国马萨诸塞州查尔斯敦第十三街149号[11] 美国纽约州纽约，纪念斯隆凯特琳癌症中心医学物理系[12] 慕尼黑工业大学医院、慕尼黑工业大学医学院与健康学院、伊萨尔河右岸医院诊断与介入放射科[13] 慕尼黑工业大学医院、慕尼黑工业大学医学院与健康学院、德国心脏中心心血管放射学与核医学科

摘要

人工智能（AI）有望通过自动化图像分析并加速临床研究，推动医学影像领域的发展。然而，人工智能实现方式和架构种类繁多、文档不一致以及可复现性问题，限制了其在研究和临床中的应用。在此，我们介绍 MHub.ai，这是一个开源的基于容器的平台，通过最少的配置实现对人工智能模型的标准化访问，提升医学影像领域的可访问性和可复现性。MHub.ai 将同行评审出版物中的模型打包为标准化容器，支持直接处理 DICOM 和其他格式，提供统一应用程序接口，并嵌入结构化元数据。每个模型都附带公开可用的参考数据，可用于确认模型是否正常运行。MHub.ai 初步收录了一系列适用于不同成像模态的先进分割、预测和特征提取模型。该模块化框架支持调整任何模型，并支持社区贡献。我们通过肺部分割模型的比较评估，在一个临床应用案例中展示了该平台的实用性。为进一步提升透明度和可复现性，我们公开发布生成的分割结果和评估指标，并提供交互式仪表板，方便读者查看个案并复现或扩展我们的分析。通过简化模型使用流程，MHub.ai 支持使用相同的执行命令和标准化输出进行并列基准测试，并降低临床转化的门槛。

引言

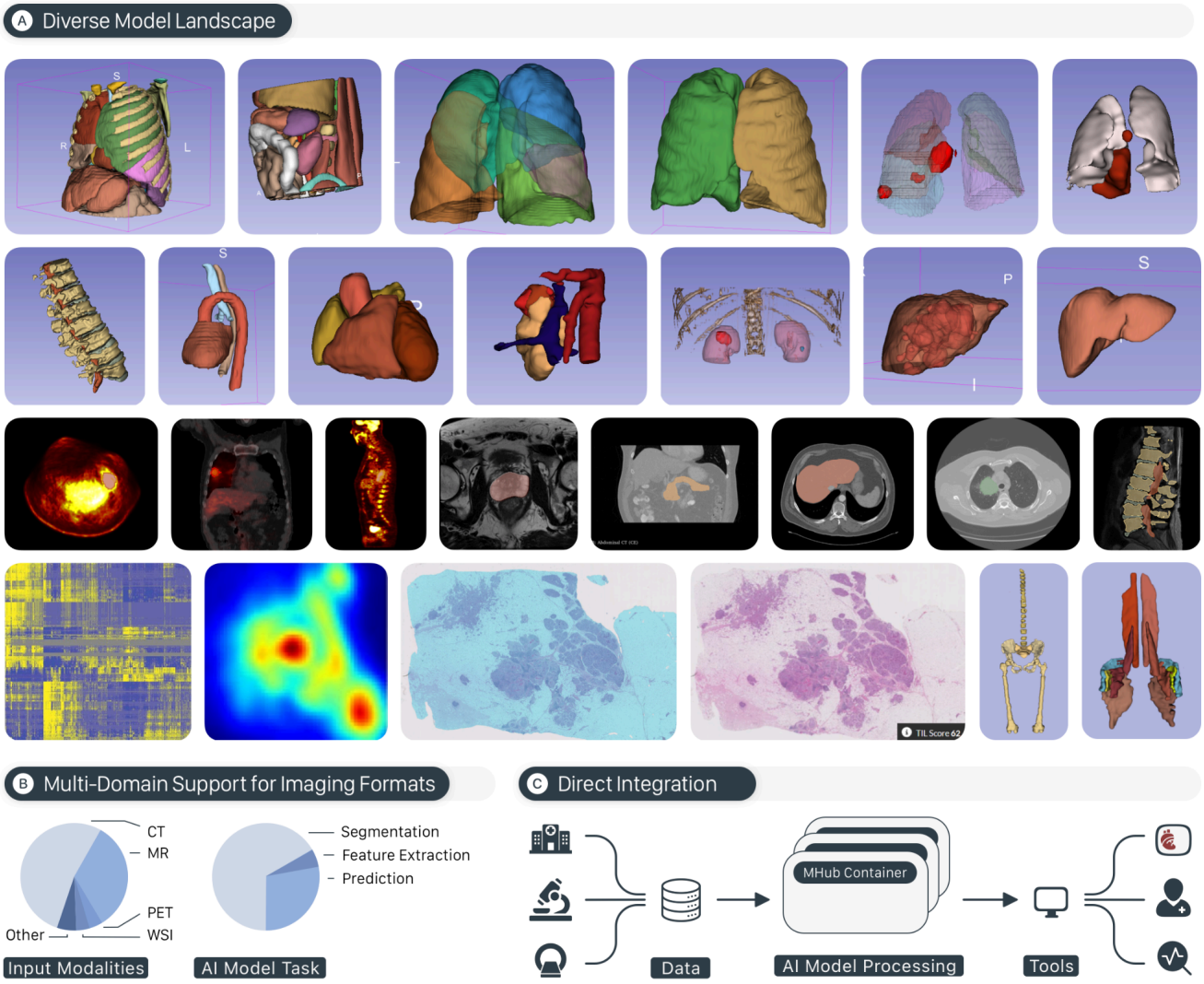


图 1 a) 多元的模型版图：MHub.ai 平台中涵盖的模型代表性示例，涉及多种成像模态、领域和任务。超过半数的模型在一个或多个输出方面与其他模型重叠，从而提供了更多选择。b) 支持多种成像格式：该框架支持多种输入模态，例如计算机断层扫描（CT）、磁共振成像（MRI）、正电子发射断层成像（PET）、全切片成像（WSI）、X 射线，以及联合图像专家组（JPG）或便携式网络图形（PNG）等图像格式，并且可扩展以支持其他成像模态和文件格式。该框架支持分割、预测和特征提取模型，这些模型均收录于模型仓库中。c) 直接集成：MHub.ai 框架将现有 AI 流水线适配为临床影像标准格式 DICOM，从而能够直接与现有的公共和私有数据提供方，以及用于可视化、检查、整理和分析的第三方工具集成。

人工智能（AI）方法（如卷积神经网络）在解读放射学、病理学及其他领域的医学影像方面展现出非凡能力（Hosny 等，2018；Niazi 等，2019；Huynh 等，2020；Kann 等，2021；McGenity 等，2024；Mienye 等，2025）。多项研究表明，这些模型能够执行分割、分类等任务，并且

其特征提取准确度可与人类专家相媲美（Liu 等，2019；Aggarwal 等，2021）。然而，要将这些进展应用于实际临床和科学工作流程，仍面临诸多挑战（Korfiatis 等，2025；Xu 等，2025；Bontempi 等，2024；Seneviratne 等，2019）。

主要障碍包括模型实现不一致、安装流程复杂、严格的数据预处理要求、配置参数不明确以及任意的输出格式——而不完整或不充分的文档往往会进一步加剧这些挑战（Koçak 等，2024；Theriault-Lauzier 等，2024；Beam 等，2020）。此外，尽管医院和诊所使用的文件格式标准是 DICOM（Clunie 等，2000；Mildenberger 等，2002），人工智能模型通常在其他格式上进行训练和运行，因此需要手动进行转换和组织，这一过程往往耗时较长。综合而言，这些因素给其他研究人员造成了很大阻碍，妨碍了模型的可比性，并削弱了可复现性。因此，大多数已发表的人工智能模型仍未被外部研究人员使用，也很少被转化应用于临床实践（Seneviratne 等，2019；Najjar，2023）。

本文介绍 MHub.ai，这是一个为医学影像提供容器化人工智能模型的平台，配备标准化应用接口、内置 DICOM 兼容性以及可复现性测试。该模型仓库提供了种类丰富的公共医学影像人工智能模型，适用于不同的输入模态、影像领域和任务（见 **图 1a、b**）。底层框架提供了统一不同模型应用接口的工具，并通过采用现有标准，实现与现有数据提供方、数据组织和可视化工具的直接集成（**图 1c**）。通过统一人工智能模型推理并提供自动化可复现性测试，MHub.ai 减少了模型执行的技术障碍，使第三方研究人员能够持续一致地使用、评估和比较模型，从而支持系统性基准测试并降低临床应用的门槛。

首先，我们介绍平台架构和框架，随后概述模型仓库，该仓库提供可直接使用的公开模型。通过使用公开临床数据对比平台上公开提供的三个肺分割模型，展示 MHub.ai 的实用性。

结果

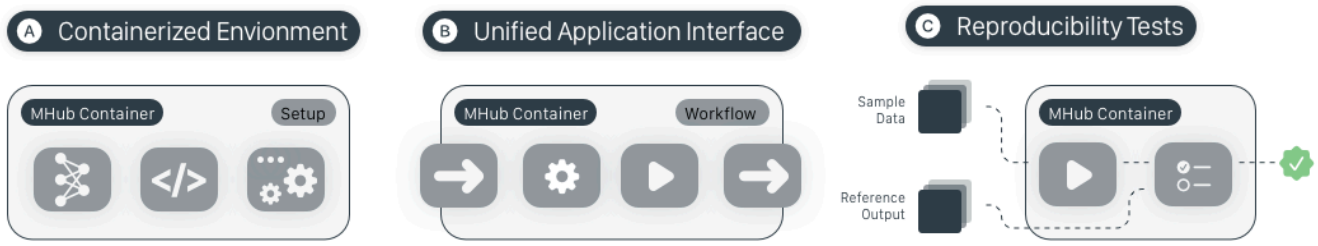


图 2 *a)* 容器化模型。每个人工智能模型都封装在一个独立容器中，其中包含推理代码、预训练权重、依赖项、库和运行环境，可实现与平台无关的执行，无需单独设置。*b)* 统一应用程序接口。标准化接口提供原生 DICOM 处理、工作流编排，以及各模型间一致的输入和输出结构。*c)* 可复现性测试。每个模型都提供公开样本数据和参考输出。框架的测试引擎会使用特定样本数据运行各个模型，将输出与参考数据进行比较，并以便于人类阅读和机器解析的格式报告测试结果。

系统设计与架构概述： MHub.ai 是一个模块化平台，由三个主要组件构成（见**图 2**）。第一个组件是容器化环境，每个模型都基于 MHub.ai 基础镜像进行打包，并包含推理源代码、模型权重、依赖项和运行环境（**图 2a**）。容器化实现了可移植性、跨平台兼容性和隔离性，而基础层共享则减少了磁盘占用。

第二个组成部分是所有容器共用的统一应用程序接口，用于统一模型的输入和输出（IO）、配置及执行方式。所有容器均可直接接收原生 DICOM 格式及其他格式的临床图像，并按照每种输出类型的标准化表示形式提供输出（例如，分割结果采用 DICOM 分割对象，表格数据采用 JSON/CSV），同时引用输入图像唯一标识符。MHub.ai 框架通过标注和可配置的工作流协调数据流（**图 2b**）。每个模型都配有结构化元数据，其中包含对模型架构、预期用途、训练和测试数据集、性能评估及局限性的标准化文档，并以标准化的模型卡框架（Mitchell 等，2019）为基础。我们在此基础上扩展了相关定义，加入输入和输出数据、支持的输入模态、训练期间是否使用增强图像、建议的最大层厚、许可证信息、原始源代码仓库和出版物的引用，以及示例图像等描述符。

第三个组成部分是可复现性测试：每个模型都包含公开的真实示例输入和存储在 Zenodo（欧洲核子研究中心，2013）中的参考输出数据（**图 2c**）。MHub.ai 框架的测试引擎会加载并处理示例输入数据，将生成的输出与参考数据进行比较，并生成机器和人类均可读取的报告，列出上下文差异，例如缺失和多余的文件、分割的 Dice 分数，以及预测模型结构化文件的键值比较。所有 MHub.ai 模型都要在贡献流程中接受测试（**参见附录 A 的 A7 节**），并在发布到 MHub.ai 模型仓库之前再次测试。

模型仓库：我们整合了 30 个经过同行评审的开源人工智能模型，涵盖分割（n = 20）、预测（n = 8）和特征提取（n = 2），适用于计算机断层扫描（n = 18）、磁共振成像（n = 10）、正电子发射断层显像（n = 3）、X 射线（n = 1）和病理学（n {v10）等成像模态（**图 1b；附录 C，表C1**）。代表性模型包括用于全身解剖结构分割的*TotalSegmentator*（Wasserthal 等，2023）、用于心脏亚结构分割的*Platipy*（Finnegan 等，2023），以及用于器官特异性任务的*nnU-Net*系列模型（Isensee 等，2020）。在肿瘤学领域，癌症影像生物标志物基础模型可支持影像生物标志物的自监督发现（Pai 等，2024）。对于正电子发射断层显像/计算机断层扫描中的肿瘤分割，平台纳入了 AutoPET 挑战赛中的一款领先模型（Peng 等，2023）。*LungMask*可快速、准确地自动完成肺及肺叶分割（Hofmanninger 等，2020），而*PyRadiomics*则支持用于预测建模的定量特征提取（van Griethuysen 等，2017）。

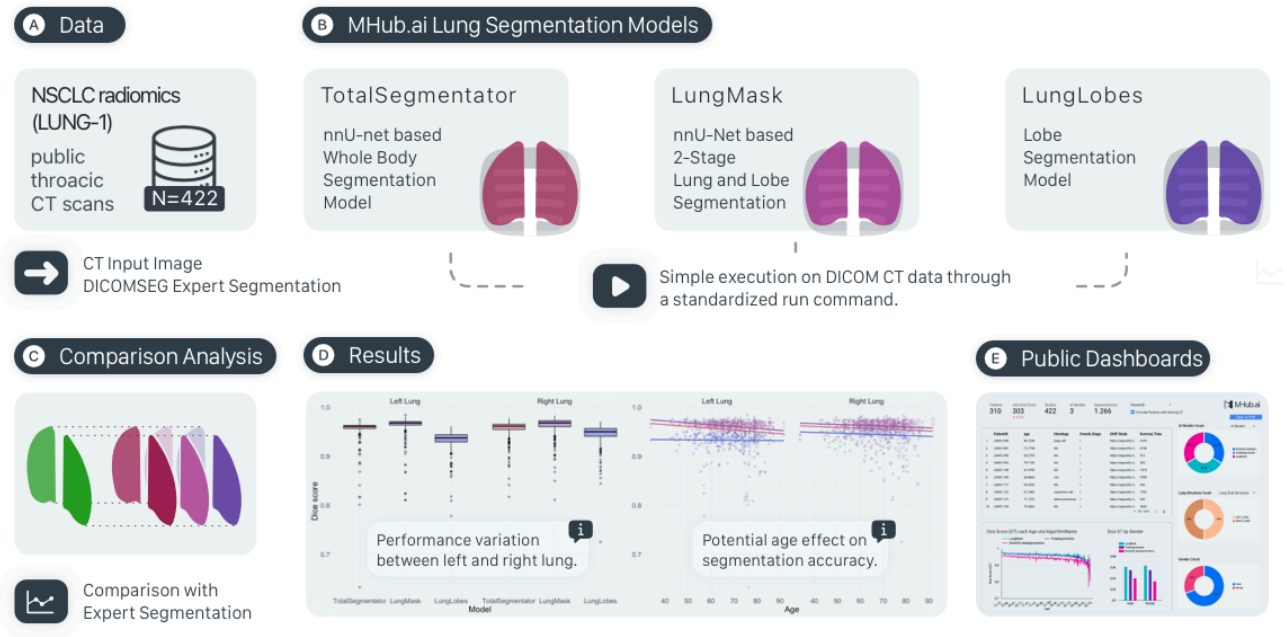


图 ***3*** 肺部分割模型的简化比较。

a) 公开数据集输入：处理了来自公开 NSCLC-Radiomics 数据集（Aerts 等，2019）的共 422 例胸部 CT 扫描，其中 303 例提供了专家分割标注。b) 统一模型执行：通过标准化的 Docker 运行命令，直接在 DICOM 文件上执行了三个分割模型，而 MHub.ai 框架

自动执行 DICOM 预处理和模型执行。c) 比较分析：使用 Dice 相似度比较三个模型的标准化分割结果与专家标注，以评估和比较不同模型在右肺和左肺上的表现。d) 模型差异洞察：使用 Dice 分数指标衡量自动模型分割结果与专家参考标注之间的一致性，并按模型展示左肺和右肺的结果（左图），以及各模型中每侧肺的结果与年龄的关系（右图）。两种视图中的 Dice 分数均较高，且模型的相对排名稳定。e) 公共仪表板：结果以 交互式仪表板的形式公开，供用户重现和扩展我们的分析结果，并完全通过网页浏览器进行可视化病例级检查。

应用案例：肺分割模型比较： MHub.ai 目前包含三个独立开发的肺分割模型：TotalSegmentator（Wasserthal 等，2023年）、LungLobes（Xie 等，2020）和 LungMask（Hofmanninger 等，2020）（见 **图 3b**）。此应用案例展示了 MHub.ai 如何支持对针对同一临床任务而独立开发的人工智能模型进行标准化、可复现的比较（**图 3**）。

我们在公开的 NSCLC-Radiomics 数据集（Aerts 等，2019）中选取 n = 422 次胸部 CT 扫描评估了这三个模型，其中 n = 303 位患者有专家提供的左肺和右肺标注。所有模型均使用相同的 MHub.ai 执行命令直接处理 DICOM 文件（**附录 C，图 C1**）。MHub.ai 框架自动完成 DICOM 预处理、模型执行，并整理生成的分割结果。为与专家参考标注相匹配，我们将肺叶级输出合并为左肺和右肺掩膜（**附录 C**）。

使用每个模型与专家标注之间的 Dice 相似度评估分割性能，并分别计算左肺和右肺的结果。所有模型均取得了较高的总体 Dice 分数。在某些分层分析中发现了虽小但具有统计学显著性的差异：LungLobes 模型对左肺的准确度略低（$\Delta = -0.0115$，$p < .001$；参见 **附录 C，表 C2**）。在 LungMask（相关性 =-0.132，$p < .01$）和 TotalSegmentator（相关性 = -0.116，$p < .01$）中，观察到 Dice 分数准确度与年龄之间存在较弱的负相关（参见 **附录 C，表 C3**）。男性和女性患者之间未发现具有临床相关性的差异（**附录 C，表C1**）。完整的统计结果见 **附录 C**。

所有分割输出及其衍生指标均可通过交互式仪表板公开获取（**图 3e；附录 C，图 C4**），从而支持队列级摘要，以及对个别扫描和分割结果进行病例级检查。

MHub.ai 支持直接在公开的临床 DICOM 数据上执行，提供统一的应用程序接口并生成兼容 DICOM 的输出，从而减少了跨多个人工智能模型执行、比较、检查和复现分析所需的工作量。

与外部工具集成：模型实现、输入、输出和元数据的标准化也支持与第三方系统的无缝集成。通过实现 MHub.ai

平台接口并采用现有文件标准，所有符合所提议格式的模型都可立即使用（见**图 1c**）。我们开发了 MHubRunner（https://github.com/MHubAI/SlicerMHubRunner/），这是面向开源应用 3D Slicer 的扩展（Pieper 等，2005 年）。该扩展使用我们的公共基于网络的 API 获取 MHub.ai 模型列表，并为用户提供图形用户界面（GUI），用于探索和运行人工智能模型、以可视化方式整合结果，并将其直接应用于临床审阅工作流。此外，标准化的分割输出以及对原始镜像的引用，使其能够在 OHIF（Ziegler 等，2020）等基于网络的查看器中直接使用，从而支持在现有影像环境中的灵活可视化与集成。

讨论

MHub.ai 为医学影像中的人工智能模型推理提供了一个标准化框架。尽管该领域已有许多公开的人工智能模型，但由于实现方式各不相同、环境设置复杂、支持和文档有限，以及可复现性不确定，这些模型很少被第三方广泛采用。MHub.ai 通过结合模型容器化、标准化应用接口、对临床医学成像数据格式的原生支持、结构化模型元数据以及自动化可复现性检查，为训练好的人工智能模型定义了一种一致的分发和执行格式，从而降低了在临床成像数据上应用和比较模型的门槛。

要让第三方能够重复使用人工智能模型，这些模型必须易于发现、设置和执行。对于模型开发和训练，TensorFlow（Martín Abadi 等，2015年）、PyTorch（Paszke 等，2017）、nnU-Net（Isensee 等，2020）和 MONAI（Cardoso 等，2022）等广泛采用的框架在很大程度上解决了这一问题；这些框架提供了标准化的抽象和接口。然而，在推理时，仍缺乏系统层面的方法：模型在打包方式、输入图像的提供格式、推理调用方式以及结果返回方式上各不相同。虽然有些模型提供容器化发行版，但并非总能以开箱即用的镜像形式获取，而且对临床数据格式和输出的支持仍不一致。MONAI 模型动物园、Hugging Face（Hugging Face，n.d.）和 BioImage 模型动物园（Ouyang 等，2022）等现有模型仓库集中展示模型，便于用户发现模型，但并未强制采用标准化执行接口、统一的数据处理方式，或结构化且完整的元数据和文档。一个值得注意的例子是 TotalSegmentator（Wasserthal 等，2023），这是一个被外部研究项目使用的独立人工智能模型，它安装简便，并为多种分割任务提供一致的推理接口。不过，这些特性目前尚未形成系统性方案，仍仅适用于个别模型。

要在临床或研究工作流中实施人工智能模型，必须评估其可靠性。人工智能模型的准确度通常以在内部或外部数据集上测得的指标来报告。

然而，评估集在规模和人口统计特征方面的差异，使得仅凭报告的指标难以比较人工智能模型。此外，这些指标对于人工智能模型将要实施的特定应用案例究竟有多大代表性，也值得商榷。此前已有研究建议，应改为在目标人群中递归地进行内部评估（Youssef 等，2023）。因此，使用同一数据集评估人工智能模型执行同一任务的表现，可以在相同且（视数据可用性而定）可控的条件下，直接比较模型输出。在 CT 扫描上进行肺分割就是这样一项任务。准确地分割 CT 扫描中的肺部，对于诊断、定量成像和放射治疗计划至关重要；精确勾画肺部可确保有效给药，同时避免损伤健康组织（Carmo 等，2022，Hofmanninger 等，2020）。在我们的示例应用案例评估中，我们展示了 MHub.ai 模型的标准化接口如何实现模型的简便直接比较，以及与公开临床数据的集成。

MHub.ai 在推理时强制采用标准化结构来打包和分发医学影像人工智能模型。模型贡献者必须定义明确的端到端工作流，提供结构化、机器可读的元数据，包含具有代表性的示例数据，并通过自动化可复现性测试，模型方可纳入公共模型仓库。这种设计确保模型以自包含的可执行软件包形式交付，具有一致的应用程序接口，并原生支持临床影像标准，默认采用 DICOM。因此，集成到 MHub.ai 中的模型无需复杂的单独设置，也无需耗时的手动数据转换或整理，即可直接应用于临床数据。此外，标准化的输出有助于模型更一致地集成到研究和临床评估工作流中。

所提出的方法和设计决策也给平台当前的状态带来了一些局限。首先，MHub.ai 模型目前使用 Docker 容器化，我们预先构建的容器要求主机系统安装 Docker 引擎。虽然 Docker 得到广泛应用，并提供了简单易用的桌面应用程序（Docker Desktop），但安装时需要根权限，且可能受到某些机构政策的限制。MHub.ai 框架独立于容器化方法，其底层原则并不局限于 Docker，也可以应用于其他容器运行时，例如 Singularity（Kurtzer 等，2017）。其次，尽管遵循我们的格式和指南、满足广泛的元数据要求并通过可复现性测试有诸多益处，但模型开发者或第三方贡献者在提交模型时仍需投入额外精力。为此，我们提供了 18 个可复用模块，以减少集成工作，并配有循序渐进的教程和详尽的文档。此外，MHub.ai 独立于其他现有打包方法运行，也可与之并行使用，从而减少供应商锁定，并借助我们的存储库扩大模型的覆盖范围。第三，目前的支持尚未涵盖所有成像模态（例如，TIFF 以外的全切片病理图像

），这限制了其在某些情境中的适用性。得益于平台模块化且可扩展的架构，今后可以补充这方面的支持。第四，可复现性测试基于选定的样本数据集，这些数据集可能无法代表临床异质性的全部范围，因此可能限制结果的普适性。第五，大多数模型需要图形处理器，要么是硬性要求，要么是因为运行速度差异显著，这对主机基础设施构成了限制。

尽管所提出的工作为此提供了稳定的基础，但 MHub.ai 是一个开放平台，专门设计为可扩展，以支持未来的改进与发展。将格式支持扩展到多个平台（包括 Singularity 等容器运行时）将提升与不同计算环境的兼容性。通过嵌入式评分模块集成性能指标的自动量化，将简化端到端基准测试。通过 GitHub 贡献实现社区驱动的扩展（例如针对病理学模块的领域特定扩展），将扩大 MHub.ai 框架所提供的可复用工具覆盖范围。总体而言，这些发展将进一步降低采用门槛，提高可复现性，并支持对医学影像领域人工智能模型进行更可靠、更具可扩展性的验证。

方法

我们介绍了人工智能模型打包策略（见 **图 4**），其中包括 MHub.ai 框架、元数据、可复现性测试、用于支持平台贡献的 CI/CD流水线，以及针对该应用案例的数据集和模型评估。

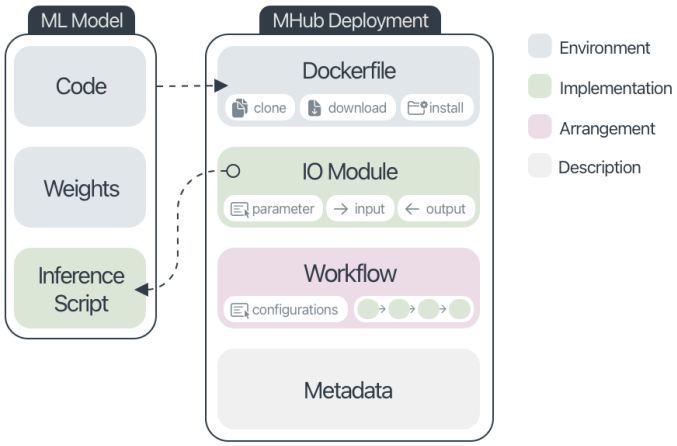


图 4 MHub.ai模型部署格式。

MHub.ai 部署模板分为环境、实现、编排和描述。系统依赖项、模型推理代码和训练权重打包在由 Dockerfile 定义的容器中。推理脚本通过 IO 模块适配器集成，其中包括所需输入数据的标注、输出数据的描述以及可配置参数。DICOM 到 DICOM 的执行流程由工作流文件编排，其中定义了可配置参数以及 IO 模块的顺序执行。模型元数据单独提供，并采用已定义的模式。

容器化

所有模型均使用 Docker 容器化。所有模型均在 Docker Hub 上公开提供预构建、可直接使用的容器镜像（仓库：mhubai/model-name）。我们定义了 Dockerfile 结构，其中包含以下步骤：以 `mhubai/base:latest` 为基础，安装模型依赖项，安装模型源代码，下载模型权重，加入 MHub.ai 封装程序，并设置 MHub.ai 默认入口点（参见**图 4**中的"环境"）。我们提供预装并配置了常用工具和框架的基础镜像。基础镜像依赖于主机系统上可用的 CUDA GPU 驱动程序，这些驱动程序必须与可用硬件匹配，因此我们有意不将其包含在基础镜像中。更多详情请参见**附录 B1**。

MHub.ai 框架

MHub.ai 运行环境以 Python 包的形式实现。模块化和可配置性是核心概念。每项任务都由一个 IO 模块实现（参见 IO 模块，**图 4**），并对应一个 IO 模块类。框架提供 Python 装饰器，方便标注配置参数以及输入和输出数据，这些内容会在运行时自动解析。框架包含 18 个基础模块，用于支持医学图像处理：导入器会搜索输入目录，并将相关文件导入统一的语义结构；筛选器会根据实例属性或文件可用性有选择地优化数据集；转换器可在 NIFTI、MHA、TIFF、PNG、DICOMSEG 和 RTStruct 等格式之间转换图像；导出器会汇总并输出元数据和自定义报告；整理器则负责协调输出内容的选择与格式。人工智能推理代码由自定义 IO 模块调用（参见实现，**图 4)**），该模块封装原始人工智能模型，并定义可配置参数以及输入和输出数据。IO 模块被组织成工作流，由运行时引擎执行。工作流包含标签、说明、IO 模块的执行顺序，以及单个 YAML 文件中的配置参数（参见工作流，**图 4**）。对于每个 MHub.ai 模型，必需的默认工作流都会处理 DICOM 输入，并在适用情况下生成 DICOM 输出（例如，所有分割模型）。更多详细信息请参见**附录 B2-9**。

元数据模式

模型的元数据包含在单个 meta.json 文件中（参见"元数据"部分和 **图 4**）。我们在代码仓库中公开提供了一个 JSON 架构（Pezoa 等，2016），用于定义该文件的结构、必需和可选键以及值类型，并将其作为持续集成流水线的一部分自动验证。更多详细信息请参见 **附录 A4**。

可复现性测试

每个模型都为每个工作流包含一个示例输入，即专门为该模型选定的代表性输入。只要条件允许，我们就使用来自公开数据集的真实示例图像。模型在示例图像上运行以生成参考输出，该输出与示例输入一起存储在 Zenodo（欧洲核子研究中心，2013）上。作为 MHub.ai 框架的一部分，我们为每个模型容器提供一个比较工具，用于加载示例数据和参考数据、运行模型，并将所有结果输出为 YAML报告。比较检查在目录级别、文件级别和内容级别进行：目录级别列出缺失和新增文件；文件级别记录文件大小差异；内容级别则根据文件类型进行检查。对于分割结果，如果与参考分割相比 Dice 分数低于 0.99，则报告为偏差。对于 JSON 或 CSV 等结构文件，则报告缺失和新增的键及值差异。该报告会创建为机器和人类均可读的 YAML 文件。作为贡献流程的一部分，我们会对每个模型执行独立的可复现性检查。相关结果在 GitHub 上公开。可复现性测试不限于所提供的示例输出和参考输出，还可用于确保研究和外部项目的可复现性。更多详细信息请参见 **附录 B10**。

持续集成/持续交付

持续集成和持续部署（持续集成/持续交付）通过 GitHub 自动化实现，可确保每个新模型或更新都经过自动格式验证和元数据验证。我们设置了一条自动部署流水线，可根据拉取请求（PR）创建模型、运行测试流水线，并将测试报告更新到拉取请求中。更多详情请参阅 **附录 A7**。

数据集与评估

我们的分析纳入了来自公开的非小细胞肺癌影像组学数据集（Aerts 等，2019）的 n = 422 名非小细胞肺癌（NSCLC）患者。每位患者均有肺部 CT 扫描，其中 n = 303 名患者提供了左肺和右肺的独立专家分割。我们使用 Python 实用工具 idc-index（Fedorov 等，2023），从 Imaging Data Commons（IDC）存档下载了患者队列的 DICOM 文件。我们使用默认运行命令，直接在下载的数据上运行了 MHub.ai 上提供的三种肺分割模型（TotalSegmentator、LungMask、LungLobes）。对于生成肺叶级分割的模型，我们将其输出汇总为左肺和右肺的复合肺掩膜，以便使用 MHub.ai 框架提供的标准化语义标签与专家标注进行直接比较。LungMask 模型已能分别提供全肺分割，无需额外处理。Dice

相似度通过对模型生成的分割结果与专家分割进行配对比较来计算。更多详情请参阅**附录 C**。

软件可用性

- 代码、工作流和容器可在 https://github.com/MHubAI 获取。
- 网页界面、模型元数据和文档可在 https://mhub.ai/ 获取。
- 3D Slicer 插件（MHubRunner）可通过 Slicer 扩展管理器获取，也可在 https://github.com/MHubAI/SlicerMHubRunner。
- 可通过以下链接查看包含肺部分割结果、Dice 分数和临床数据的交互式仪表板 https://lookerstudio.google.com/reporting/0c405365-31f5-4714-8b2a-03631e7a0686。

附录 A - 平台设计



图 A1 弥合临床实施鸿沟的要求，旨在实现人工智能模型的集成与使用。

A1 - 弥合实施鸿沟

文献已充分记录人工智能模型在临床实践和医学研究中的潜力（Hosny 等，2018；Niazi 等，2019；Huynh 等，2020；Kann 等，2021；McGenity 等，2024；Mienye 等，2025）。然而，在知名期刊上发表并以宽松许可证开放的模型中，只有少数成功转化为临床应用，或被研究实验室采用。造成这一实施鸿沟的因素有很多。既往研究指出，可操作性、患者安全和实用性不足是主要问题（Seneviratne 等，2019），这些问题可以概括为对人工智能模型普遍缺乏信任。人工智能的整合通常可视为对新技术的一项投资。人们对人工智能模型的信任往往仅与（患者）安全相关，并与隐私、模型准确性和稳定性、模型泛化能力及人群偏倚等问题相联系；但要考虑进行此类投资，还需要有信心（即相信会取得积极成果）（见**图 A1**）。尽管多项研究指出人工智能模型有降低成本的潜力（Molwitz 等，2025；van Leeuwen 等，2022），但实施人工智能模型的成本往往仍不明朗。开源模型和开放许可证模型会在宽松许可证下提供完整的源代码和训练后的模型权重，且无需支付许可费用即可自由使用。一些期刊要求这样做，以提高透明度并强调开放获取，同时也使外部研究部门和机构在技术层面能够使用这些模型。然而在实际操作中，实施此类模型仍需投入大量时间、技能和经验，而且通常无法保证成功。

因此，还应关注在医院或研究部门设置人工智能模型所涉及的挑战和成本。尽管需要考虑额外投资，例如信息技术基础设施、存储以及计算单元（如配备 GPU 的机器）的投入，或聘请人工智能专家的费用，但这些通常是与整体人工智能投资相关的一次性支出，而非针对单个人工智能模型的支出。本文聚焦于技术工作量和相关挑战，这些因素使得人工智能模型难以设置并成功融入临床工作流程或研究流程。我们提出一种解决方案，以减少这类工作量，从而间接降低集成人工智能模型的成本，并帮助弥合实施鸿沟。

A2 - 模型设置和应用程序式模型打包

人工智能模型可以远程设置，例如使用云基础设施（Bontempi 等，2023），也可以在本地环境中设置。设置过程包括安装模型及其依赖项，例如系统库、所需的软件工具和预训练模型权重。该过程的复杂程度、可能遇到的障碍以及总体耗时，会因各个模型的具体要求、本地或远程信息技术基础设施的局限性和复杂性、用户的经验以及模型文档的质量而有很大差异。

人工智能模型通常使用 Python 等脚本语言编写，而非以编译后的可执行文件形式发布，并且依赖特定的系统库和第三方工具（依赖项）。要使用这些模型，必须确保系统上具备所有依赖项（例如，已安装这些依赖项）。依赖项版本解析、依赖项安装、文件路径设置、文件提供和系统配置统称为环境设置。由于平台差异、依赖项冲突以及库缺失或过时，即使是经验丰富的开发者，也往往难以重现完全相同的环境设置。

一种广泛采用的解决方案是容器化。容器化运行环境（容器）提供了一个沙箱环境，与主机系统环境隔离，并专为人工智能模型设置。容器的蓝图以静态、不可变的镜像形式提供。环境设置以一组命令的形式定义在*Dockerfile*。 从干净的系统环境（例如特定版本的 Ubuntu）开始，Dockerfile 中的命令会在构建过程中按顺序执行，以生成镜像。在此过程中，人工智能模型会与所有必需的依赖项一同配置和安装。生成的镜像易于共享，例如作为压缩文件，或上传到镜像仓库（如 Docker Hub）。与复杂且依赖于系统的设置过程相比，执行镜像只要求在主机系统上安装并运行容器引擎。

一些人工智能模型已经提供了 Dockerfile，列出环境设置的步骤，并给出构建镜像的蓝图。虽然这是向前迈出的一步，但从 Dockerfile 构建镜像本身并不能保证成功。构建过程中仍可能出现依赖项冲突和问题，例如由于任何依赖项（或其依赖项的依赖项，……）的补丁和更新所致。为了提供独立完整的环境，镜像通常有数 GB 大。Docker 镜像通过为每条命令创建一层，并在镜像之间共享相同的初始层，来缓解这一问题。因此，如果两个镜像基于相同的基础镜像，并且对共享依赖项采用相同的命令顺序，就可以减少总磁盘占用。不过，要同步 Dockerfile，就需要制定所有开发者都遵循的标准（例如，提供统一基础镜像）。此外，容器内的环境设置尚无标准，包括容器中预期的输入文件位置，以及生成的输出文件存放位置。人工智能模型还常常不将大型模型权重作为代码的一部分，而是在首次执行时下载。对人工智能模型进行容器化时，这种做法会带来两个问题。首先，它会导致运行时必须连接互联网。其次，由于容器总是从镜像中定义的全新环境启动，每次容器启动时都会重新下载，从而增加执行时间。因此，应避免依赖在线依赖项，而应在构建过程中下载大型文件，并将其作为镜像的一部分一同交付，使其完全独立。

比较通用模型与容器化模型的安装要求（见**图 A2**）可以发现，通过容器化，人工智能模型环境设置中的各项准备步骤被整合到模型镜像中，安装要求也简化为容器引擎（例如 Docker Desktop）。

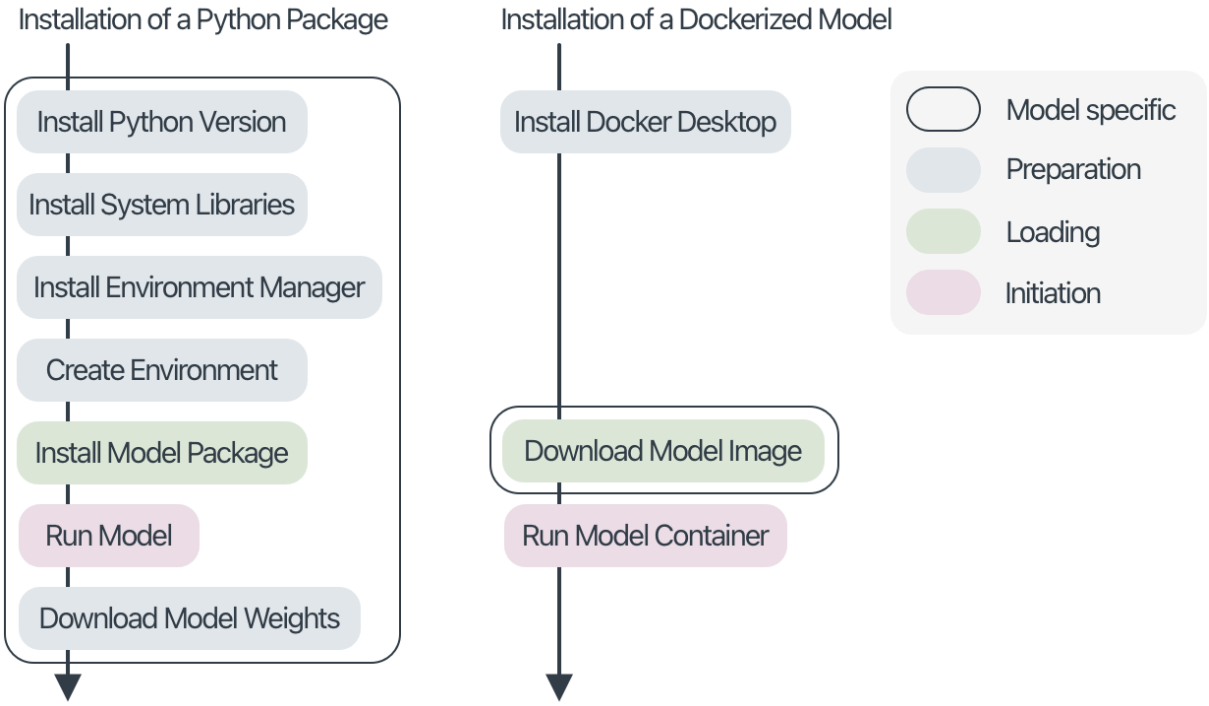


图 A2 比较以 Python 包形式提供的通用人工智能模型与容器化模型在准备、加载和启动方面的简便程度。带边框的模块因模型而异，必须根据模型版本进行调整。容器化减少了模型特定的安装步骤。

MHub.ai 采用容器化来部署人工智能模型，并提供一个基础镜像，其中包含相关工具以及供所有模型共用的框架（见**附录 B，B1 节**）。

A3 - 模型协调统一

将人工智能模型打包到容器中，可大幅简化设置和安装。复杂且因模型而异的环境设置，只需一次配置容器引擎即可完成。然而，各模型的应用程序接口，以及模型执行前（输入）和执行后（输出）对数据准备的特定要求，仍存在显著差异。这导致不同模型的用户体验参差不齐，数据组织也需要重复投入精力，且这项投入仍需针对每个模型单独进行（见**图 A3**）。

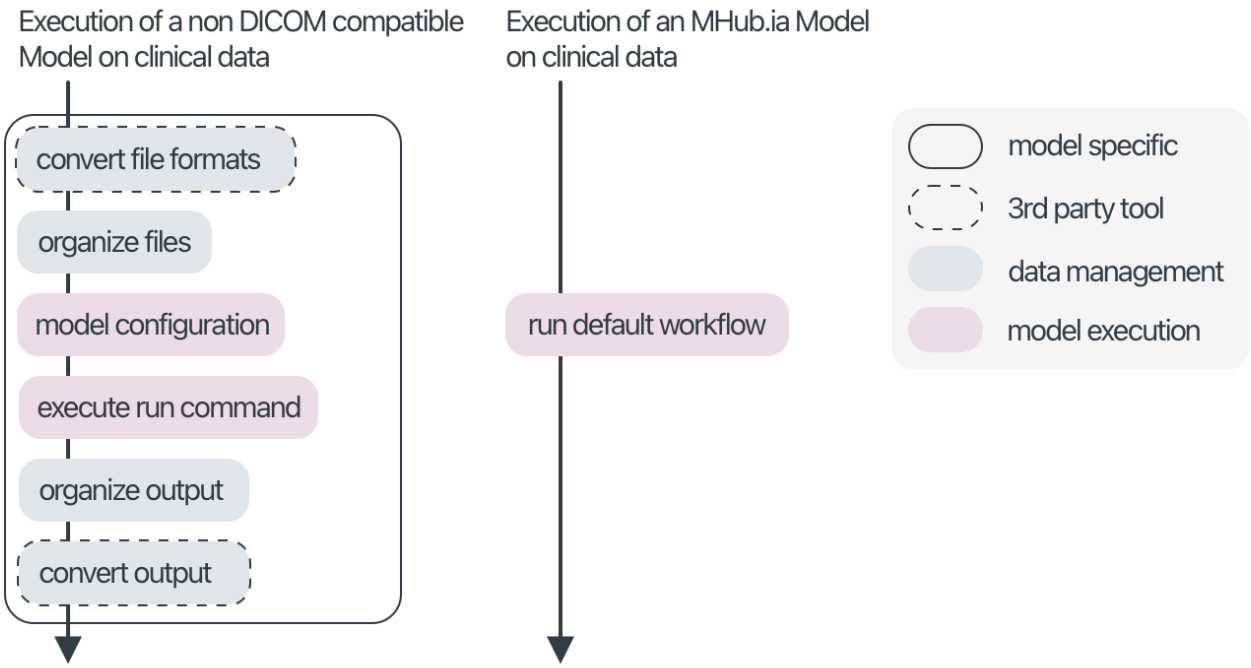


图 A3 对比通过应用程序接口调用以 Python 包形式提供的通用人工智能模型与容器化 MHub.ai 模型的交互方式：当输入数据为 DICOM 格式的临床图像，且人工智能模型默认不支持 DICOM 输入时，前者需要借助第三方工具定制文件转换，手动将文件整理为所需的输入格式，并手动整理输出，以便随后将其转换为兼容的 DICOM 格式。相比之下，在适用的情况下，MHub.ai 模型默认直接处理 DICOM 数据并返回 DICOM 格式，无需针对各个模型进行任何专门调整。

为解决这一问题，我们开发了 MHub.ai 框架（**附录 B，B2 节**），以统一模型的**输入**、**输出**、**配置**和**执行**。通过为所有 MHub.ai 模型提供统一标准的应用程序接口，这些模型可以轻松集成到现有工作流中，支持用类似模型替换模型，并能以合理的工作量进行本地评估和模型比较。

1993 年，医学影像领域引入了医学数字成像和通信（DICOM）标准（Clunie 等，2000 年；Mildenberger 等，2002 年）。然而，这种数据格式并不适合训练人工智能模型，因此许多开发者会先将数据转换为更合适的格式。因此，大多数模型都要求特定的输入格式（通常是训练时所用的格式），而临床实际使用的则是 DICOM 标准。我们提供的转换模块随基础镜像一起交付，并与人工智能模型捆绑。因此，所有 MHub.ai 模型都具有**统一的输入接口**，并可直接处理 DICOM 数据。

人工智能模型输出计算结果的格式，通常也由模型创建者任意定义，既包括语义层面（例如如何解读输出，即 100 percent = 1），也包括结构层面（例如以何种格式呈现输出，即 0 = background、1 = heart）。将人工智能模型集成到现有工作流的过程中，需要同时考虑这两个方面。对于分割模型，放射治疗结构集和 DICOM 分割对象是两种扩展 DICOM标准、并关联到原始输入图像的格式。前者主要用于放射治疗中的危及器官勾画和肿瘤勾画，但在表示复杂形状方面存在局限（Shrestha 等，2024）。DICOM 分割对象是一种较新的格式，支持将感兴趣区域嵌入语义本体

中。对于生成预测、分类或特征提取结果的深度学习模型，最常见的文件格式是 JSON（在开发环境中更为常见）和 CSV （通常用于诊所和数据分析）。在我们的 MHub.ai 框架中，我们实现了对输出数据的抽象标注，并提供了将分割结果转换为 DICOMSEG 或 RTStruct 文件的模块，以及生成临床数据 JSON 或 CSV 报告的模块。输出格式的选择和规范在工作流中定义，从而实现**统一的输出接口**。默认工作流始终为分割模型生成 DICOMSEG。

默认情况下，所有 MHub.ai 模型都采用基于 DICOM 的工作流。对于更高级的场景，可提供不同的输入格式、非 DICOM 输出、适配本地文件结构以及其他定制功能，作为替代工作流。除了输入和输出数据的规范外，人工智能模型流水线的行为通常还受其他参数控制。这些参数的设置方式由开发者决定，且因模型而异。在 MHub.ai 模型中，可配置参数以标准化的方式公开（参见 **附录 B，图 B3**），并作为工作流配置的一部分进行指定。虽然默认的零设置工作流简单且标准化，但也可提供自定义配置作为替代工作流。综合这些特性，可实现**一致的执行接口和配置**。

A4—元数据与仓库

选择人工智能模型时，明确界定并说明其预期用途、要求和限制非常重要，这有助于避免出现意外行为。这包括有关可接受输入数据的信息，例如模型接受的成像模态（计算机断层扫描、MR 等）；有关输出语义的信息，以便正确解读模型结果；以及对其他参数和配置选项的说明。在前文各节中，我们已介绍如何为模型设置、输入、输出、配置和执行提供标准化接口。本节将进一步介绍一种标准化的模型元数据格式。

Mitchell 等人（2019）提出了模型卡框架，以标准化有关模型架构、预期用途、训练数据和测试数据、性能评估以及已知算法局限性的文档。我们在此定义基础上扩展了可选内容，例如输入和输出数据的图像及描述符，其中包括支持的输入模态、与增强图像的兼容性，以及建议的最大层厚。我们还纳入了相关信息，例如模型的许可证、训练后的模型权重、模型原始源代码和出版物的链接，以及原始论文的引文。每个模型的元数据均包含在单个 ` meta.json` 文件中。我们在公共代码仓库中提供了 JSON 架构（Pezoa 等，2016），以确保结构一致。。

基于现在全面、标准化且机器可读的元数据，我们可以通过两种方式提升模型的可发现性（见**图 A1**）和（执行前的）可比较性。首先，我们开发了一个公开网站和模型仓库（https://mhub.ai/models），其中包含所有 MHub.ai 模型，并提供搜索和筛选功能，用户可以直接根据关键参数筛选模型，例如输入模态，或输出分割中的特定感兴趣区域。这样，用户就可以快速搜索“*所有可在 CT 扫描上运行并勾画肺左上叶的模型*”。其次，我们以一致的方式展示每个模型页面上的元数据，使信息易于检索，也便于并排比较一个或多个模型。

A5 - 可复现性

人工智能模型环境和流水线的复杂性，不仅使得在第三方系统上重建设置变得困难，也会因依赖快速演进的第三方库和工具（依赖项），而危及人工智能模型的可复现性，例如数据管理和机器学习框架。这些第三方库和工具在快速发展的领域中积极开发和调整，可能会在后续版本中引入已知或未知的变化，同时带来改进和新功能。即使遵循相同的安装步骤，今天安装的设置未来也可能无法运行，除非为每个软件包按相同顺序安装相同版本——而这一点有时连原始开发者也无法复现，因为开发过程中经常会添加和移除软件包。一种解决方案是将每项依赖项固定到确切版本。然而，只有在每项依赖项也将其所有依赖项固定到确切版本时，这种方法才有效；但这无法保证，而且往往不受模型开发者控制。此外，这种做法甚至未必可取。例如，许多开发者会将依赖项固定到较小的版本范围，并允许进行补丁更新以获取安全相关更新。虽然这是一种普遍接受并遵循的惯例，但未来某项依赖项的一次意外更新，就可能导致整个 AI 流水线的输出失效或发生变化。

如前文所述，在特定情况下，容器化可以解决这一问题。例如，容器镜像一旦构建完成，便是整个人工智能模型流水线及所有相关依赖项的冻结快照。然而，一旦再次构建镜像，上述情况便会出现，模型、其依赖项、系统环境或任何其他未仔细固定的组件都可能发生变化。

由于可复现性对于建立信任至关重要，因此模型的可复现性是 MHub.ai 设计与开发过程中的一项重要考量。我们希望为所有模型提供一致的应用程序接口，持续改进 MHub.ai 框架，并向平台添加新模型，因此永久冻结模型并不现实。相反，我们在所有 MHub.ai 模型中实施了一种机制，以确保在镜像重新构建后，模型输出保持一致（参见 **附录 B，B10 节**）。

所有 MHub.ai 模型都会在贡献流程中接受测试（参见 **A7 节**），并在发布到 MHub.ai 模型仓库之前，于每次构建前依次进行测试（参见 **附录 D**）。

A6 - 文档与教程

在上述章节中，我们说明了将人工智能模型迁移至 MHub.ai 格式如何简化这些模型的设置与执行，以及 MHub.ai 模型仓库如何借助结构化元数据，便于发现和比较模型。为推动所提格式的采用，帮助研究人员和临床医生运行 MHub.ai 模型并将 MHub.ai 集成到新的或现有的工作流中，同时帮助开发者将自己的模型或第三方模型转换为所提格式，我们提供了详尽的文档，介绍其底层概念、如何执行和自定义现有工作流以及创建自定义工作流、MHub.ai 框架的详细技术说明、现有模块概览，以及开发者创建自定义模块的指南。

此外，我们还制作了四篇深入教程，逐步讲解如何在 Imaging Data Commons (IDC) 平台的公开 DICOM 数据上运行示例模型（在 *IDC* 集合上运行 *TotalSegmentator*、**表 A1**），如何在非 DICOM 格式的胸部 CT 扫描上运行自定义肺部分割工作流（运行自定义 *MHub* 肺部工作流

在 *NIFTI* 格式胸部 *CT* 上运行分割工作流、**表 A1**），如何以 MHub.ai 格式部署开源人工智能模型（为 *MHub* 创建阈值模型、**表 A1**），以及如何将心脏分割模型的输出从 DICOMSEG 切换为 RTStruct（以 *RTStruct* 格式运行心脏结构分割、**表 A1**）。大多数教程都包含“太长不看”（TL;DR）部分，用几条命令概述整个教程，以突出我们平台的简便性。

标题	时长	太长不看
在 IDC 集合上运行 TotalSegmentator	30 分钟	6 条命令
在 NIFTI 格式的胸部 CT 上运行自定义 MHub 肺部分割工作流	45 分钟	8 条命令
为 MHub 创建阈值模型	90 分钟	N/A
以 RTStruct 格式运行心脏结构分割	15 分钟	10 条命令

表 A1 概述：MHub.ai 平台使用、适配和扩展教程。

A7 - 社区贡献

医学影像领域的人工智能正在快速发展，新工具、框架和训练好的模型不断涌现。医学影像的复杂性，以及其涉及的各种文件格式、转换工具和标准，都要求从业者具备丰富的知识和经验。MHub.ai 不仅致力于简化人工智能的应用，还旨在为开发者提供必要的指导、工具和支持，帮助他们以易于获取、用途广泛且易于集成的格式提供自己的模型，从而扩大其工作的影响力。

我们的 MHub.ai 框架以 Python 编写，已预装在基础镜像中，是所有 MHub.ai 模型背后的驱动引擎。它提供语义数据标注与查询、灵活的工作流执行、自动化测试，以及将工作流步骤模块化为构建模块（IO 模块）的功能。它还包含 18 个 IO 模块，可用于整理数据并将其转换为不同格式。每个模型都可以提供自定义 IO 模块，例如用于模型引用和可选处理步骤的模块。

我们深知技术在迅速发展，人工智能工程师使用的框架可能在短短几年内就会过时。因此，MHub.ai 旨在独立于任何框架，不干预模型训练和开发过程。相反，我们专注于模型推理，并提供一致的接口和简洁流畅的用户体验。MHub.ai 提供了可重复执行的医学图像准备工具，但人工智能模型开发可以独立于 MHub.ai 进行。模型经过训练和评估后，即可集成到 MHub.ai 格式中。需要注意的是，这种集成并非排他性的。MHub.ai 不必是模型分发的唯一格式或平台，我们也支持在其他平台和格式上并行分发。

MHub.ai 目前通过向我们的模型仓库提交 GitHub 拉取请求来接受模型贡献。我们已实施 GitHub 自动化，可自动验证 Dockerfile、JSON 模式及其他相关组件。这些自动化还会生成一份全面的报告，其中包含指向文档相关章节的链接。

附录 B - 平台实现

在上一章中，我们介绍了当前面临的挑战，并阐述了设计决策的依据。本章将全面概述所提出的 MHub.ai 模型，以及我们开发的工具和模板。

我们定义了四个层（见 **图 4**）：环境层、实现层、编排层和元数据描述层。环境层包含模型设置及相关依赖项。该层涵盖模型代码、训练权重、依赖项、系统库和配置。实现层提供人工智能模型与 MHub.ai 框架之间的适配器。在此定义模型的可配置参数、所需输入数据和生成的输出数据。编排层用于将人工智能模型嵌入 DICOM 到 DICOM 工作流中。该工作流由一系列按顺序排列的 IO 模块组成，用于导入、转换和准备数据，执行模型，以及转换、关联和组织输出。此外，工作流中还定义了可配置的参数覆盖。每个 MHub.ai 模型都有一个默认工作流，该工作流会默认执行。模型元数据在描述层中定义。

B1 - 容器化模板

为了支持并指导模型的容器化，同时保持 MHub.ai 模型之间共享的结构，我们提供了一个预安装工具且仅需进行最少设置的**基础镜像**，以及一个用于构建 MHub.ai 模型镜像的**模板**。

基础镜像包含 MHub.ai 框架（请参阅 **B3 节**）以及常用的工具和库。我们还在此定义了用于输入和输出数据的通用文件系统端点。

MHub.ai 模型镜像的模板规定了一组用于构建 MHub.ai 模型镜像的强制性规则，这些规则适用于 Dockerfile：

> 1. 所有 Dockerfile 都必须从最新的 MHub.ai 基础镜像开始：FROM mhubai/base:latest 2. 安装并设置其他系统库、工具和依赖项。3. 必须使用提供的构建工具导入模型定义：ARG MHUB_MODELS_REPO RUN buildutils/import_mhub_model.sh my_model ${MHUB_MODELS_REPO} 4. 可以使用提供的构建工具导入其他集合：RUN buildutils/import_mhubio_collection.sh collection_name 5. 克隆模型的公开源代码或安装模型。6. 下载模型权重。7. 入口点必须指向 MHub.ai 运行脚本（mhub.run），并且默认情况下应执行默认工作流。ENTRYPOINT ["mhub.run"]CMD ["--workflow", "default"]

统一所有模型镜像可提高容器内的透明度。这也是提供统一应用程序接口的一项基本设计决策。构建指令的正确设置会在以下情况下自动进行验证：

作为我们使用 GitHub 自动化实现的持续集成与持续开发流水线（持续集成/持续交付）的一部分，在 MHubAI/models 仓库中创建新的拉取请求（PR）。

尽管其他容器化平台也是可行的替代方案，但我们选择先将工作重点放在 Docker 上，因为它安装过程简单，且提供了易于使用的图形用户界面（GUI）。不过，所述结构也可以转换为其他容器化平台。现有工具可实现向其他平台（例如 Apptainer（https://apptainer.org/））的（半）自动转换。

B2 - MHub.ai 框架

我们的框架以 Python 包的形式实现（https://github.com/MHubAI/mhubio），并预装在基础镜像中。它提供 IO 模块（参见**B3 节**）和工作流执行（参见**B6 节**）所需的核心功能，并为模型执行（参见**B7 节**）和模型可复现性测试（参见**B10 节**）提供统一的应用程序接口。

B3 - IO 模块

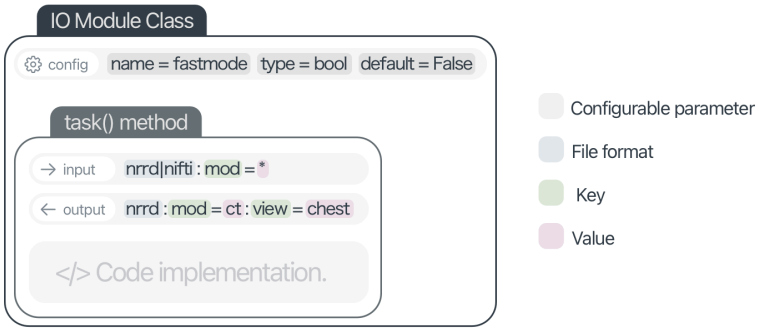


图 B1 用于可配置参数和语义输入/输出规范的声明式接口。该图展示了 IO 模块如何使用框架提供的装饰器声明可配置参数、输入描述符和输出描述符。语义描述符用于查询输入文件和定义输出文件，并在运行时解析为绝对文件路径。

MHub.ai 的一个核心概念，是将消耗输入（I）、生成输出（O）和可配置（C）的任务或功能拆分为可独立运行的 IO 模块（参见 **图 B1**）。IO 模块以 Python 类的形式实现。该框架提供了一个抽象基类，具体的 IO 模块通过扩展此类实现。可配置参数通过类变量实现。其对外公开的名称、数据类型、默认值和描述，均使用框架提供的 Python 装饰器进行标注。IO 模块类的 task() 方法包含模块实现，并由运行时引擎执行。所需的输入文件和生成的输出文件作为 task() 方法的参数声明，并使用 Python 装饰器对输入和输出进行语义描述（参见 **章节 B4**）。配置、输入和输出变量可在 task 方法中使用，并由运行时引擎自动解析。task 方法既可以包含复杂的 Python 实现，也可以采用最简单的形式运行一条命令行界面命令，传入已解析的文件路径和配置，充当简单的适配器。

B4 - 通过语义注释和查询实现路径抽象

IO 模块之间的数据交换由框架的数据管理层负责。对于每个 IO 模块，所需输入和生成的输出都通过语义查询和描述符进行声明。语义描述符会向运行时引擎注册模块的输出，并包含一个（文件）类型声明，后接描述性键值对。语义查询使用相同的语法，但可以包含运算符和占位符。在 IO 模块内部，每个输入、输出和可配置参数都对应一个变量。引擎会在运行时将这些变量解析为绝对文件路径，并在本地图中跟踪相关元数据。引擎会在图中搜索与指定类型和元数据匹配的文件，将解析出的文件作为模块输入，并为已声明的输出生成绝对文件路径。这一机制可在容器内建立一致的目录结构。语义声明使 IO 模块不受自定义命名约定和目录布局的影响，保持通用性和独立性。易读的描述符便于检查和理解数据流；所有中间文件仍可访问，并可通过在工作流配置中指定其语义描述符来导出。

B5 - IO 模块工具箱

我们的框架提供 18 个 IO 模块，分为七个功能类别，以便支持医学影像工作流：导入器可将 DICOM 等各种数据格式的导入标准化为统一的语义结构；筛选器可依据实例属性或文件是否可用，有选择地筛选数据集；转换器可在不更改底层内容的情况下，对 NIFTI、MHA、TIFF、PNG、DICOMSEG 和 RTStruct 等格式之间的影像数据进行转换；处理器可在保留数据类型的同时提取或修改数据，通常执行分割提取和配准；运行器可执行 AI 流水线以生成新的分析数据；导出器可汇总并输出元数据和自定义报告；整理器则可通过灵活的复制和删除操作管理文件的整理与清理。这些流程所需的外部工具，例如 dicomsort（Pieper，n.d.）、plastimatch（Sharp 等，2010）、dcm2niix（Li 等，2016）和 itkimage2segimage（Herz等，2017），均已预安装在我们的基础镜像中。这些模块共同支持适用于医学影像研究和临床应用的灵活、透明且可复现的数据处理流水线。

B6 - 工作流

在容器内，MHub.ai 运行时引擎会依次执行 IO 模块。它们的顺序和配置参数定义了一个工作流，该工作流由单个 YAML 文件表示。每个模型都必须包含一个默认工作流；若未另行指定，则会选择该工作流（参见 **B7 节**）。模型提供的其他预配置工作流数量不受限制。自定义工作流可以作为字符串传入运行命令的标准输入。**图 B2**展示了一个默认工作流示例。第一个 IO 模块是一个核心模块，它会扫描输入目录中的 DICOM 文件，并将其拆分为多个实例。随后，可用实例会逐一传入转换模块，此处使用的是 DICOM 到 NIFTI 转换器。完成此步骤后，每个输入实例都会对应生成一个 NIFTI 文件。接下来的 IO 模块是自定义推理适配器，用于运行封装后的 AI 流水线。模型要求输入格式为 NIFTI，并通过语义数据标注声明这一要求。作为输出，它定义了一个

NIFTI 格式的分割文件，以及分割后的 ROI（参见 **B9 节**）。引擎会对每个可用实例运行 ML Pipeline 模块，并将相应的 NIFTI 文件作为输入。所有实例处理完毕后，另一个核心 IO 模块会将生成的 NIFTI 分割结果转换为兼容 DICOM 的输出。如图中的箭头所示，该模块因此需要两个输入：DICOM 格式的原始图像和 NIFTI 格式的分割结果，并生成一个 DICOMSEG 输出文件。所有实例处理完毕后，系统会运行导出模块，将文件整理成所需的输出结构。默认情况下，系统会为每个实例在输出目录中创建一个以 SeriesInstanceUID 命名的子文件夹，并在其中放置分割文件。

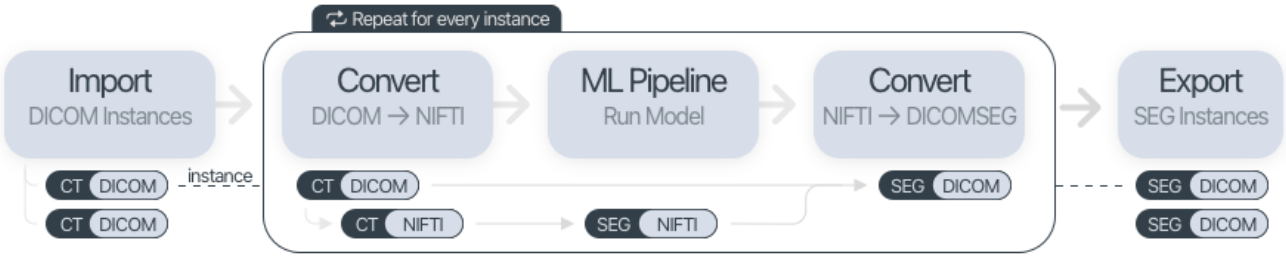


图 B2 典型 MHub.ai DICOM 到 DICOM 工作流示意图。

此示例流程从 DICOM 导入模块开始，该模块将分散的文件整理为独立实例。接下来，转换模块会将图像转换为所需的输出格式。ML Pipeline 作为自定义模块实现，用于运行算法。随后，第二个转换模块会将 Pipeline 生成的输出以及原始输入图像转换为 DICOMseg 格式。导入模块识别出的每个实例都会重复这一流程，并由导出模块将其整理到输出文件结构中。

B7 - 执行与 CLI 接口

所有 MHub.ai 模型共用相同的应用程序接口。docker run 命令会根据 MHub.ai 模型镜像创建容器，并挂载输入和输出目录。由于 MHub.ai 模型在容器内的输入和输出挂载点均已标准化（参见 **B1 节**），运行命令仅在模型名称上有所不同（参见 **图 B3a**）。在容器内，会执行 MHub.ai 框架提供的运行引擎（mhub.run）（**图 B3b**）。运行引擎提供 CLI 接口，用于指定工作流（--workflow default）、覆盖可配置参数（--config:module.parameter=value）、启用输出日志记录和调试。所有 CLI 参数均为可选项；若未指定其他工作流，则执行默认工作流。随后，运行引擎按照工作流文件中的定义，依次执行 MHub.ai IO 模块（**图 B2c**）。已部署人工智能模型的推理管线（**图 B3d**）由包装 IO 模块（m.Run）调用，并获取所需数据。

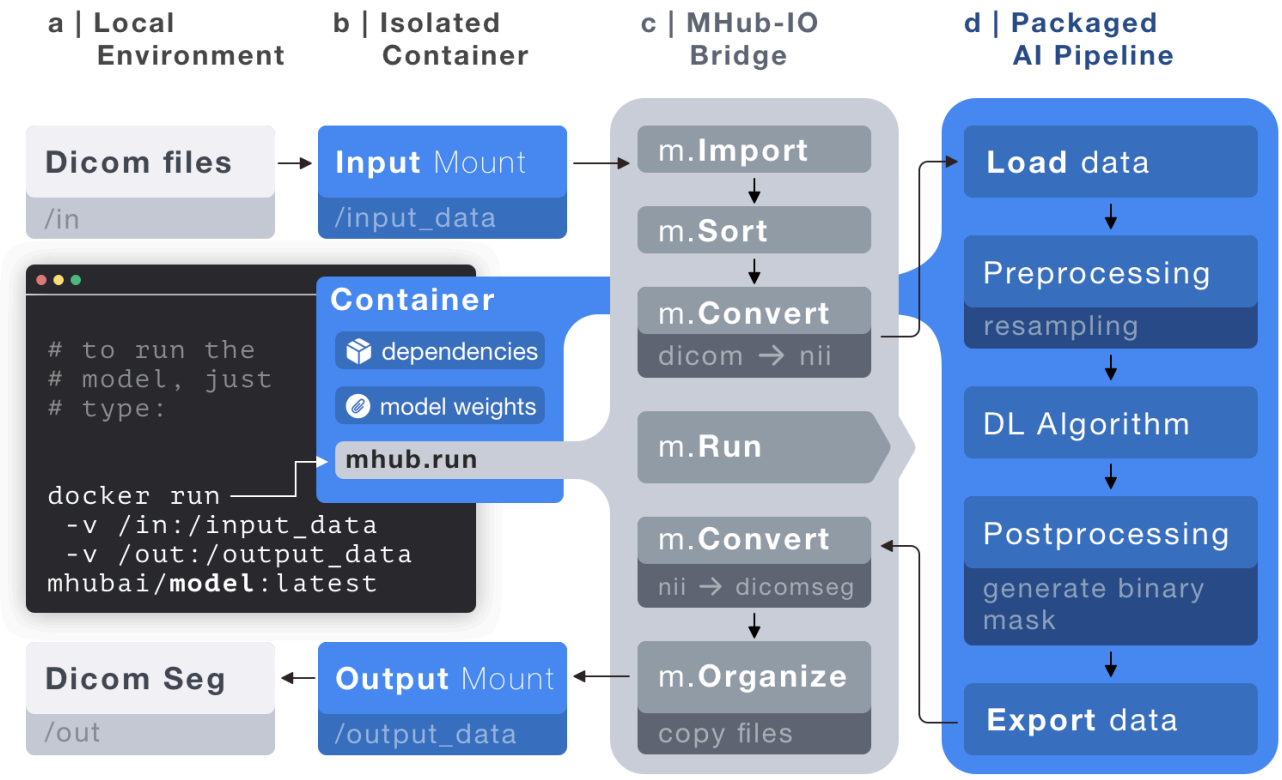


图 B3 位于 MHub.ai 容器内。

黑色箭头表示数据流经四个层级的路径。**a)** 本地环境：用户指定一个包含 DICOM 数据的输入目录，以及一个用于存储结果的输出目录。。**b)** 隔离容器：Docker 容器封装依赖项、模型权重和原始 AI 流水线，并通过标准化的 docker run 命令运行，该命令会挂载输入和输出目录。**c)** *MHub.ai IO* 粘合层：MHub.ai 执行框架负责协调各模块，以导入、排序、转换、运行和重组数据。语义数据类型和声明式接口使模块能够独立于文件系统运行。。**d)** 打包的 *AI* 流水线：原始 AI 流水线在 MHub.ai 框架完成数据转换和准备后执行。

B8 - 扩展 API

工作流并不局限于内置的核心 IO 模块。自定义模块使用与标准模块相同的接口和装饰器约定。自定义模块可以直接作为模型定义的一部分。这是通过实现一个充当适配器的包装 IO 模块来调用模型推理的方式，该模块使 MHub.ai 运行引擎能够在工作流中调用推理。此类自定义模块仅适用于随其一同提供的特定模型。对于可重复使用的 IO 模块，框架提供了扩展 API，可在构建时纳入来自第三方代码库的 IO 模块。加载工作流时，框架会自动发现并注册所有可用模块（核心模块、模型特定模块以及通过已安装扩展包提供的模块），以便在工作流配置文件中按名称引用它们。扩展包必须遵循规定的目录和模块结构；也支持来自未列出代码库的扩展包，但会触发警告。公共模型仓库中提供了官方支持的扩展包列表。通过此机制，工作流能够纳入特定领域或特定模型的处理步骤，

支持重复使用通用组件，并允许在不修改底层框架的情况下扩展功能。

B9 - SegDB

为分割模型提供兼容 DICOM 的输出，一个难点在于需要精确的标注。为了帮助开发者统一分割标签，我们提供了一个基于关键词的查找表。我们为每个感兴趣区域分配唯一标识符，例如 HEART 或 LEFT_UPPER_LUNG_LOBE，并将其映射到易于理解的名称、RGB颜色以及语义标注，其中包括类别、类型和可选修饰符。对于肿瘤、囊肿或钙化斑块等解剖学发现，语义标注的结构有所不同，需要指定上下文结构，例如解剖学发现所关联或嵌入其中的感兴趣区域。我们开发了一种直观的标注格式，例如 LEFT_LUNG+TUMOR，用于表示左肺中存在肿瘤，并在运行时自动将其解析为生成 DICOMSEG文件所需的语义标注结构。我们将此功能打包为一个公开的 Python 包（https://github.com/MHubAI/SegDB/），并已预装在我们的基础镜像中。

B10 - 测试框架

该框架还包含一个专用测试引擎，可帮助验证工作流是否能产生可复现的输出，例如在第三方系统上执行时。每个模型都提供可公开访问的样例数据和参考数据（例如通过 Zenodo 提供），以用于此目的。测试引擎会下载这些数据集，在样例输入上执行工作流，并将生成的输出与相应的参考输出进行比较。比较结果汇总在一份人类和机器均可读取的 YAML 报告中，其中记录了缺失或多余的文件、文件大小差异，以及支持的文件格式中内容层面的差异。对于分割文件，引擎会计算各分区的 Dice相似性评分，并在报告中将低于 .99 的评分标记为相对于参考分割的偏差。对于 JSON 或 CSV 等结构化格式，报告会逐个键值对列出缺失或多余的键以及不同的值。该流程仅用于评估生成的输出与参考输出之间是否等效，不会评估超出这些比较结果所能推断范围的算法性能。实际上，测试引擎有助于检测由环境差异、依赖项变更或实现错误导致的偏差；有助于在科学研究中开展透明且可共享的可复现性工作流；允许用户在不执行模型的情况下查看预期输出；提供标准化的比较方法，避免含糊不清的人工检查；并确保模型加入公共模型仓库前经过一致的质量控制。

附录 C - 肺分割应用案例

我们使用同一个临床队列，对集成到 MHub.ai 的三个公开可用的肺分割模型进行了基准测试：TotalSegmentator（Wasserthal 等，2023年）、LungLobes（Xie 等，2020）和 LungMask（Hofmanninger 等，2020）。首先，我们介绍标准化的模型执行过程（**图 C1**），随后说明队列选择和预处理。接着，使用 Dice 相似性指标评估定量性能（**图 C2b**），并进一步按人口统计变量进行分层（**图 C3**）。最后，我们介绍如何通过交互式仪表板公开我们的结果（**图 C4**）。

C1 - 数据集与模型特征

评估队列包括来自公开 NSCLC-Radiomics 数据集（Aerts 等，2019）的 422 名非小细胞肺癌（NSCLC）患者。该数据集包含每位患者的肺部 CT 扫描，以及肺部专家分割结果。其中，309 名患者提供了专家标注，分别以左肺和右肺的独立分割结果呈现。其余 113 名患者的专家标注则涵盖全肺，未区分左右侧。

所有三个分割模型均生成五个肺叶的肺叶级分割结果（左肺的上叶和下叶，以及右肺的上叶、中叶和下叶）。LungMask 使用两阶段分割流程：第一阶段生成整个左肺和右肺的分割结果，这些结果与肺叶分割一并包含在分割输出中。所有模型输出中的解剖结构均采用相同的标注（参见 **附录 B，B9 节**）。

C2 - 模型执行与输出协调

所有三个肺部分割模型均使用相同的 MHub.ai 命令行界面执行。如**图 C1**所示，所有模型的执行命令、输入数据声明和输出目录指定均相同；唯一不同的参数是模型标识符。MHub.ai 框架会自动识别各个 CT 体积数据，处理 DICOM 处理与转换，在每个图像上执行模型，并将生成的分割结果整理为标准化的输出结构。模型执行前无需手动进行数据转换或重新整理。

执行后，对模型输出进行了协调，以便与专家标注直接比较。对于 TotalSegmentator 和 LungLobes，使用 MHub.ai 框架提供的标准化语义标签，将各个肺叶的分割结果合并为复合的左肺和右肺掩膜。LungMask 输出本身已处于全肺层级，因此无需进行聚合。这种协调在不改变解剖学评估目标的情况下，调和了输出粒度方面的差异。

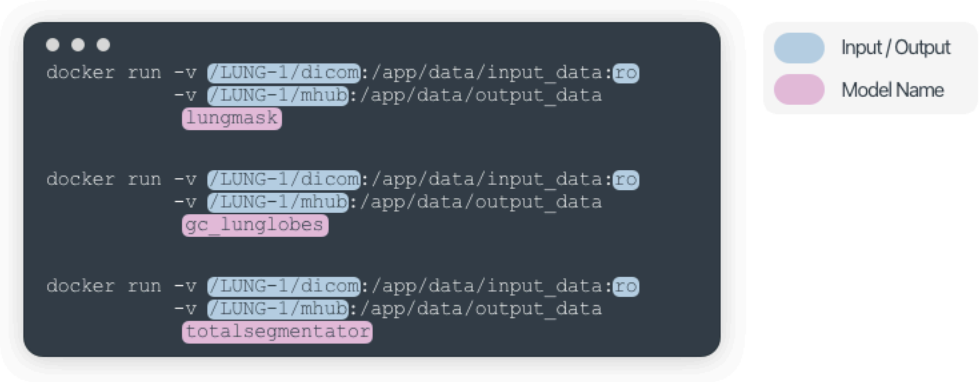


图 C1 LungMask、LungLobe 和 TotalSegmentator 模型的执行命令。所有 MHub.ai 模型的运行命令、输入数据声明和输出文件夹指定方式均相同。若要选择其他模型，只需更改运行命令中的模型名称。

C3 - 定量分析的队列排除与纳入资格

仅对专家分割结果与模型分割结果在输出协调后实现空间对齐且语义兼容的病例，使用 Dice 相似度进行了定量评估。由于模型输出与专家标注之间的图像几何不匹配，导致处理错误并无法计算重叠度，因此排除了 6 名患者。对全肺进行了专家标注的 113 名患者被排除在分析之外。其余所有病例构成了下文所报告统计分析的基础。在所有图和表中，均一致使用了由 303 名患者组成的同一筛选后的队列。

C4 - 定量评估与结果

使用 Dice 相似系数对分割准确度进行量化，该系数通过逐一比较各模型的输出与相应的专家肺分割结果计算得出。分别计算左肺和右肺的 Dice 分数。

图C2展示了按模型、肺侧和性别分层的 Dice 分数分布。三个模型的总体 Dice 分数均较高，表明其与专家标注具有较高的一致性。人口统计特征方面观察到的差异促使我们进行了进一步的统计比较。

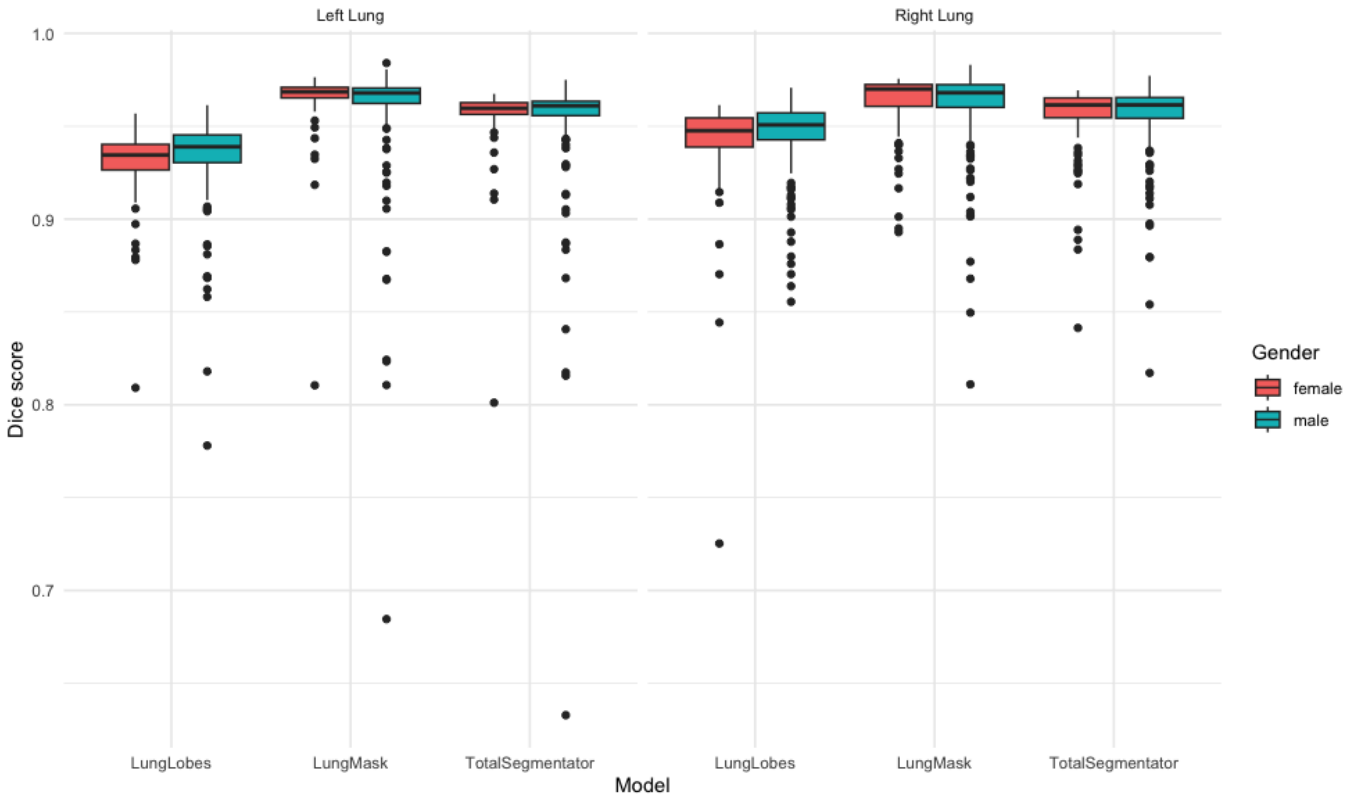


图 2 专家分割结果与模型对左肺和右肺的预测结果之间 Dice 分数的比较（范围为 0.6 至 1，未排除异常值），按模型和性别分组。

首先，我们比较了男性和女性患者中的模型准确性。我们对男性和女性患者的模型与专家标注之间的 Dice 分数进行了独立样本t检验；结果见**表C1**。LungLobes 模型在性别相关差异方面具有统计学显著性（$p < .05$，差异为 -0.00519），而 LungMask 和 TotalSegmentator 均未发现显著差异。考虑到 Dice 分数相似度的差异非常小，不到 .01 分，这种差异不太可能具有临床相关性。

模型	估计值	p
LungLobes	-0.00519	p = .0171
LungMask	0.00267	p = .155
TotalSegmentator（v1）	0.000478	p = .807

表C1 按模型比较不同性别（男性与女性）之间 Dice 分数的 t 检验。列出了每个模型的均值（估计值）差异及相应的 p 值。

在第二项实验中，我们比较了左肺和右肺之间的模型准确性。我们对三个模型进行了独立样本t检验；结果见**表 C2**。我们观察到一处轻微但

具有统计学显著性的差异：对于 LungLobes 模型，左肺和右肺的准确度存在差异。Dice 相似度对目标大小较为敏感，而由于心脏移位，左肺在解剖结构上较小，因此相同的绝对分割误差可能导致 Dice 分数出现不成比例的降低。

模型	估算	p
LungLobes	-0.0115	p <.001
LungMask	-0.000435	p = .828
TotalSegmentator（v1）	-0.00123	p = .539

表 C2 对各模型的左右肺 Dice 分数进行 t 检验比较。各模型的均值差异（估计值）及相应的 p 值如下。

Dice 分数与患者年龄之间的相关性见**表 C3**，并展示于**图 C3**。LungMask 和 TotalSegmentator 与年龄呈弱但具有统计学显著性的负相关，而 LungLobes 则未发现显著相关性。散点图显示各年龄组之间存在大量重叠，表明年龄相关效应相较于总体病例间变异较小。

模型	cor_s	p_s
LungLobes	0.0128	p = .756
LungMask	-0.132	p = .00134
TotalSegmentator（v1）	-0.116	p = .00485

表 C3 各模型的 Dice 值与年龄之间的斯皮尔曼秩相关。表中列出了斯皮尔曼 rho（相关性$_s$）及对应的 p 值（p_s）。所有模型的样本量（n = 590）均相同。

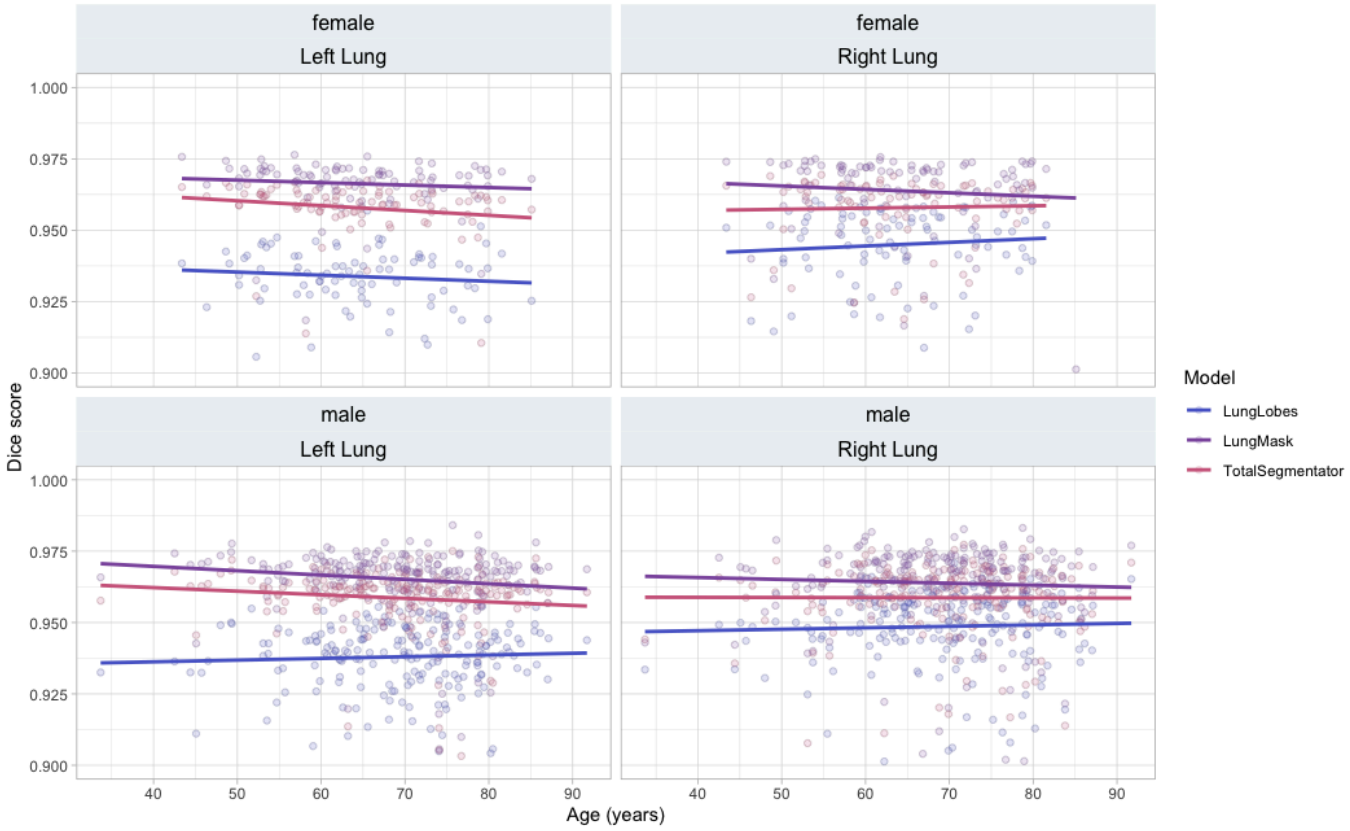


图 C3 各模型按年龄划分的分割与真实标注之间 Dice 分数（介于 0.9 和 1 之间）散点图，并按肺（左、右）和性别（男、女）分别显示。

C5 - 可复现性与交互式仪表板

为提高透明度和可复现性，分割输出及其衍生的 Dice 分数可通过交互式仪表板公开访问（**图 C4**）。这些仪表板提供队列级摘要，并支持病例级访问，可查看单个 CT 扫描及其对应的分割结果；这些内容关联至基于云端的 OHIF（Ziegler 等，2020）查看器。仪表板地址：
https://lookerstudio.google.com/reporting/0c405365-31f5-4714-8b2a-03631e7a0686。

借助仪表板，可以复现**图 C2-C3**中显示的所有汇总统计，并追溯到具体病例，从而检查异常值、失效模式和人口统计学分层。

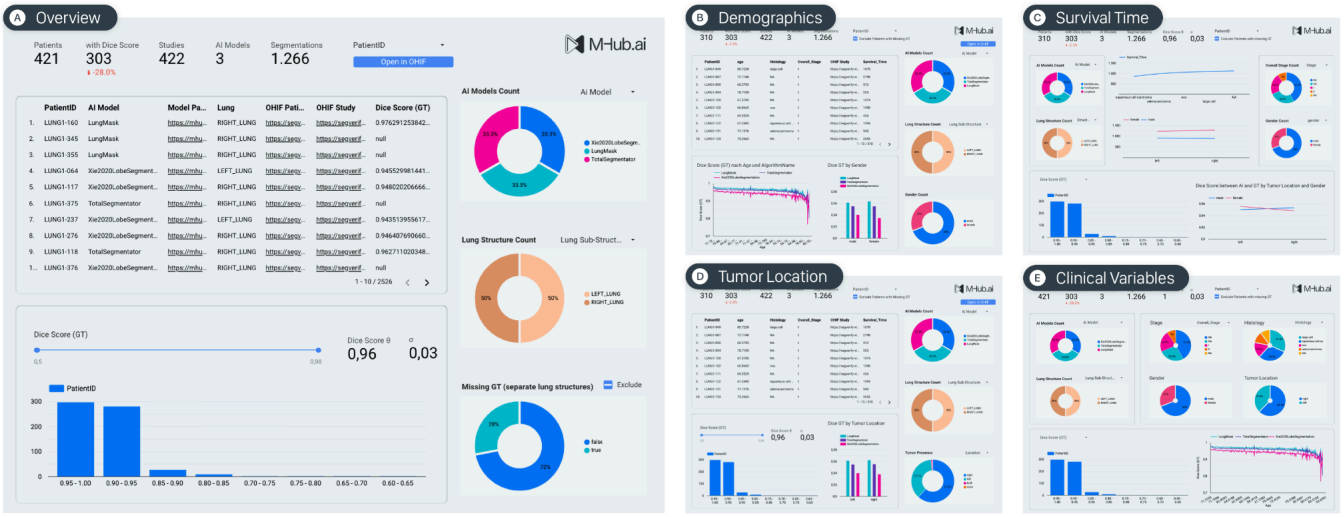


图 4 支持对肺分割模型进行透明且可复现评估的交互式仪表板。仪表板提供队列摘要、性能指标，并支持按病例查看分割结果及对应的 CT 扫描，使用户能够依据人口统计学和临床变量进行探索。A–E 面板概述了模型的总体性能（A）、按性别和年龄分组的模型性能（B）、生存时间相关分析（C）、肿瘤存在情况下模型性能的变化（D），以及按组织学和癌症总体分期分组的模型性能（E）。

附录 D - 模型仓库

所有 30 个模型标准化后的概况见**表 D1**。对于每个模型，我们收集了有关模型原始输入图像的文件格式、模型原始输出的文件格式、如何通过应用程序接口执行模型推理，以及模型用户手册（文档）的信息。

如表格所示（见**A | 原始**列），30 个人工智能模型中有 20 个要求使用非 DICOM 输入格式，因此需要手动将 DICOM 图像转换为用于推理的其他格式。五个模型实现了自定义逻辑，可从文件夹中加载 DICOM 图像，但要求该文件夹中只包含恰好对应一幅图像的文件，因此可能需要手动整理数据（例如，对 DICOM 文件进行排序和分离）。三个模型实现了自定义逻辑，可从混合 DICOM 文件中分离图像；另有一个模型仅以 MHub.ai 格式发布。

20 个分割模型中，只有 7 个能够以兼容 DICOM 的 DICOMSEG 格式提供带有语义组织标注的勾画结果。所有预测模型都采用各自独立且固定的结构化输出格式（主要为 JSON，其中一个模型还提供 CSV 作为替代格式）。

该应用程序接口是面向 11 个模型的自定义通用命令行界面（CLI）；其中五个模型基于 nn-UNet 模型，并使用相同的接口。12 个模型要求将输入和输出目录挂载到 Docker 容器中的不同位置，其中九个模型不包含额外的配置参数。一个模型需要自定义元数据文件来描述输入数据集，并提供了一个单独的脚本来创建该文件。对于某个模型，输入和输出数据及其他配置均由 MONAI 配置文件定义。

所有模型中，不到 50%（12/30）提供了文档，完整说明安装和设置流程、应用程序接口的参数，介绍可能的准备工作和后续步骤，并给出模型使用示例。九个模型提供了简要说明或不完整的示例。另有九个模型没有提供文档。

模型	T	A | 原始				B | MHub.ai			
		输入数据	输出格式	接口	手册	输入 Data	输出格式	接口	手册
TotalSegmentator（Wasserthal 等，2023年）	S	NIFTI	NIFTI	自定义命令行界面	Yes	DICOM	DICOMSEG	工作流	是
Platipy(Finnegan 等，2023年)	S	NIFTI	NIFTI	自定义命令行界面	Yes	DICOM	DICOMSEG	工作流	是
BAMF磁共振前列腺分割（Murugesan 等，2023年）	S	NIFTI / DICOM	NIFTI / DICOMSEG	Docker 挂载	Yes	DICOM	DICOMSEG	工作流	是
BAMF CT 肾脏分割(Murugesan 等，2023年)	S	DICOM	DICOMSEG	工作流	Yes	DICOM	DICOMSEG	工作流	是
CT 肝脏分割(Isensee 等，2020)	S	NIFTI	NIFTI	nn-UNet 命令行界面	是	DICOM	DICOMSEG	工作流	是
LungMask（Hofmanninger 等，2020）	S	NIFTI / DICOM	NIFTI	自定义命令行界面	Yes	DICOM	DICOMSEG	工作流	是
CaSuSt（Nürnberg 等，2022）	S	NIFTI / DICOM	NIFTI	自定义命令行界面	Yes	DICOM	DICOMSEG	工作流	是
肺叶分割(Xie 等，2020)	S	MHA / DICOM	MHA	自定义命令行界面	No	DICOM	DICOMSEG	工作流	是
胰腺分割Isensee 等人 , 2020	S	NIFTI	NIFTI	nn-UNet 命令行界面	是	DICOM	DICOMSEG	工作流	是
组织背景分割于组织病理学全切片图像(Bándi等，2019年)	S	TIFF	MHA	自定义命令行界面 + 配置文件	No	DICOM	TIFF	工作流	是
BAMF CT肝脏分割（Murugesan 等，2023年）	S	NIFTI / DICOM	NIFTI / DICOMSEG	Docker 挂载	Yes	DICOM	DICOMSEG	工作流	是
AutoPET 挑战赛第二名挑战赛：假阳性减少网络 Peng 等，2023	S	MHA	MHA	Docker 挂载	No	DICOM	DICOMSEG	工作流	是
全前列腺分割(Isensee 等，2020)	S	NIFTI	NIFTI	nn-UNet 命令行界面	是	DICOM	DICOMSEG	工作流	是
前列腺移行区和外周区分割（Isensee 等，2020）	S	NIFTI	NIFTI	nn-UNet 命令行界面	是	DICOM	DICOMSEG	工作流	是
用于基础模型的是癌症影像生物标志物（Pai 等，2024）	F	NIFTI	JSON	自定义命令行界面	Yes	DICOM	JSON	工作流	是
PI-CAI 挑战赛基线(Bosma 等，2023年)	S	MHA	MHA	Docker 挂载	No	DICOM	DICOMSEG	工作流	是
胸部危及器官 (nnU-Net)(Isensee 等，2020)	S	NIFTI	NIFTI	nn-UNet 命令行界面	是	DICOM	DICOMSEG	工作流	是
胸部 X 线片结节定位器（Node21 基线）（Sogancioglu 等，2024）	P	MHA	JSON	自定义命令行界面 + Docker 挂载	Yes	DICOM	JSON	工作流	是
计算机断层扫描 肺癌风险预测（Liao 等，2019）	P	MHA	JSON	Docker 挂载	No	DICOM	JSON	工作流	是

胰腺导管腺癌检测 计算机断层扫描 (Alves 等，2022年)	S+P	MHA	MHA + JSON	Docker 挂载	No	DICOM	DICOMSEG + JSON	工作流	是
PyRadiomics （van Griethuysen 等，2017）	F	NIFTI	JSON / CSV	自定义命令行界面 + 配置文件	Yes	DICOM	JSON	工作流	是
脊柱分割（SPIDER 基线） （Lessmann 等，2019年）	S	MHA	MHA	自定义元数据文件	Yes	DICOM	DICOMSEG	工作流	是
BAMF 磁共振肝脏分割 （Murugesan 等，2023年）	S	NIFTI / DICOM	NIFTI / DICOMSEG	Docker 挂载	No	DICOM	DICOMSEG	工作流	是
前列腺移行区和外周带分割 （Prostate158） （Adams等，2022）	S	NIFTI	NIFTI	自定义 MONAI 配置	Yes	DICOM	DICOMSEG	工作流	是
TIGER 挑战赛获胜者：VUNO 团队 （VUNO，2022）	P	TIF	JSON	Docker 挂载	No	DICOM	JSON	工作流	是
STOIC2021 基线 (Boulogne 等，2023)	P	MHA	JSON	Docker 挂载	Yes	DICOM	JSON	工作流	是
BAMF PET CT 乳腺 （Murugesan 等，2023年）	S	NIFTI / DICOM	DICOMSEG	Docker 挂载	Yes	DICOM	DICOMSEG	工作流	是
SMIT 自监督肺 GTV 分割 (Jiang 和 Veeraraghavan，2024)	S	NIFTI	NIFTI	自定义命令行界面	否	DICOM	DICOMSEG	工作流	是
氟代脱氧葡萄糖正电子发射断层扫描/计算机断层扫描肺部及 **肺肿瘤标注** (Murugesan 等，2023)	S	NIFTI / DICOM	DICOMSEG	Docker 挂载	Yes	DICOM	DICOMSEG	工作流	是
MRSegmentator （Häntze等，2024年）	S	NIFTI / MHA / NRRD / DICOM	DICOMSEG	自定义命令行界面	Yes	DICOM	DICOMSEG	工作流	是

表 D1 人工智能模型原始部署格式与 MHub.ai 部署格式的比较。绿色 = 表示已标准化 / 可访问，红色 = 表示未标准化 / 不可访问，黄色 = 表示部分标准化 / 部分可访问。模型类型（T）中，S = 表示分割，P = 表示预测，F = 表示特征提取。

将全部 30 个模型实现为 MHub.ai 格式后，标准化带来的效果在表格中实施前后的直接对比中显而易见（见**B | MHub.ai**列）。这 30 个模型均可直接在临床数据上运行，提供标准化且具有临床相关性的输入格式。24 个分割模型中的 23 个以标准化的 DICOMSEG 文件形式提供输出，文件包含一致的语义组织标注、标签和颜色代码，并附有指向原始 DICOM图像的链接。其中一个模型使用病理切片显微成像（SM），目前 DICOMSEG 尚不直接支持这种成像方式。所有 4 个预测模型均以原始的任意格式提供预测结果，并额外将其转换为语义值，以便纳入通用且完全可由工作流配置的报告。全部 30 个模型均通过遵循某种

明确且有完善文档说明的格式的工作流文件进行配置。全部 30 个模型共用相同的应用程序接口，使用相同的执行命令，并依托适用于所有 MHub.ai 模型的详细文档。此外，这 30 个模型均提供真实样本数据和参考输出，并且都通过了我们的可复现性测试。

MHub.ai 模型仓库的一个关键特征是各模型之间分割输出的重叠。分割结果以标准化输出格式提供，无需额外的数据准备或后处理即可进行直接的成对比较。在 30 个集成模型中，有 17 个（56.7%）至少与另一个模型共享一个分割输出（见 **表 D2**）。结合较低的设置和执行成本，MHub.ai 从而支持对独立开发的人工智能模型进行系统性比较和交叉评估。

输出重叠的模型数量	模型	占比
0	13	43.3%
1	5	16.7%
2	1	3.3%
3	3	10.0%
4	6	20.0%
10	1	3.3%
12	1	3.3%
≥1	17	56.7%

表 D2 MHub.ai 模型仓库中的模型级重叠。与其他模型至少共享一种分割输出的模型分布（n = 30 个模型）。

参考文献

Adams, L. C., Makowski, M. R., Engel, G., Rattunde, M., Busch, F., Asbach, P., Niehues, S. M., Vi nayahalingam, S., van Ginneken, B., Litjens, G., & Bressem, K. K. (2022). Prostate158——一个由专家标注的用于前列腺癌检测的 3T 磁共振成像数据集及算法。*Computers in Biology andMedicine*，*148*(105817)，105817。

Aerts, H. J. W. L., Wee, L., Rios Velazquez, E., Leijenaar, R. T. H., Parmar, C., Grossmann, P., Carvalho, S., Bussink, J., Monshouwer, R., Haibe-Kains, B., Rietveld, D., Hoebers, F., Rietbergen, M. M., Leemans, C. R., Dekker, A., Quackenbush, J., Gillies, R. J., & Lambin, P. (2019)。*NSCLC-Radiomics* 数据[数据集]。癌症影像档案。https://doi.org/10.7937/K9/TCIA.2015.PF0M9REI

Aggarwal, R., Sounderajah, V., Martin, G., Ting, D. S. W., Karthikesalingam, A., King, D., Ashrafian, H., & Darzi, A. (2021)。医学影像中深度学习的诊断准确度：一项系统综述与荟萃分析。*Npj Digital Medicine*，*4*(1)，65。

Alves, N.、Schuurmans, M.、Litjens, G.、Bosma, J. S.、Hermans, J. 和 Huisman, H.（2022）。用于在计算机断层扫描中检测胰腺导管腺癌的全自动深度学习框架。*Cancers*，*14*(2)，376。

Bándi, P.、Balkenhol, M.、van Ginneken, B.、van der Laak, J. 和 Litjens, G.（2019）。利用卷积神经网络在全切片组织病理图像中进行与分辨率无关的组织分割。*PeerJ*，*7*(e8242)，e8242。

Beam, A. L.、Manrai, A. K. 和 Ghassemi, M.（2020）。医疗保健中机器学习模型可复现性面临的挑战。*JAMA*：美国医学会杂志，*323*(4)，305–306。

Bontempi, D.、Nuernberg, L.、Pai, S.、Krishnaswamy, D.、Thiriveedhi, V.、Hosny, A.、Mak, R. H.、Farahani, K.、Kikinis, R.、Fedorov, A. 和 Aerts, H. J. W. L.（2024）。利用云端在放射学中实现端到端可复现的人工智能流水线。*Nature Communications*，*15*(1)，6931。

Bosma, J. S., Saha, A., Hosseinzadeh, M., Slootweg, I., de Rooij, M., & Huisman, H. (2023年). 使用报告引导的伪标签进行半监督学习，以基于深度学习的方法利用双参数磁共振成像检测前列腺癌。《放射学：人工智能》, *5*(5), e230031.

Boulogne, L. H., Lorenz, J., Kienzle, D., Schon, R., Ludwig, K., Lienhart, R., Jegou, S., Li, G., Chen, C., Wang, Q., Shi, D., Maniparambil, M., Muller, D., Mertes, S., Schroter, N., Hellmann, F., Elia, M., Dirks, I., Bossa, M. N., ... Dubois, M.-P. R. (2023年). STOIC2021 COVID-19 人工智能挑战赛：将可复用训练方法应用于私有数据。载于 *arXiv [eess.IV]*. https://doi.org/10.48550/ARXIV.2306.10484

Cardoso, M. J., Li, W., Brown, R., Ma, N., Kerfoot, E., Wang, Y., Murrey, B., Myronenko, A., Zhao, C., Yang, D., Nath, V., He, Y., Xu, Z., Hatamizadeh, A., Myronenko, A., Zhu, W., Liu, Y., Zheng, M., Tang, Y., ... Feng, A.（2022）。MONAI：一个用于医疗保健深度学习的开源框架。载于*arXiv [cs.LG]*。arXiv。http://arxiv.org/abs/2211.02701

Carmo, D., Ribeiro, J., Dertkigil, S., Appenzeller, S., Lotufo, R., & Rittner, L.（2022）。肺及其肺叶自动分割方法和公开数据集的系统综述，以及计算机断层扫描图像中的发现。《医学信息学年鉴》，*31*(1)，277–295。

Clunie, D. A. (2000 年). *DICOM*结构化报告. PixelMed Publishing.

欧洲核子研究组织, & OpenAIRE. (2013). *Zenodo*. 欧洲核子研究中心. https://doi.org/10.25495/7GXK-RD71

Fedorov, A., Longabaugh, W. J. R., Pot, D., Clunie, D. A., Pieper, S. D., Gibbs, D. L., Bridge, C., Herrmann, M. D., Homeyer, A., Lewis, R., Aerts, H. J. W. L., Krishnaswamy, D., Thiriveedhi, V. K., Ciausu, C., Schacherer, D. P., Bontempi, D., Pihl, T., Wagner, U., Farahani, K., ... Kikinis, R. (2023年). 美国国家癌症研究所影像数据共享平台：迈向医学影像人工智能的透明度、可复现性和可扩展性。《放射学图谱》：北美放射学会评论出版物, *43*(12), e230180.

Finnegan, R. N., Chin, V., Chlap, P., Haidar, A., Otton, J., Dowling, J., Thwaites, D. I., Vinod, S. K., Delaney, G. P., & Holloway, L.（2023年）。开源、全自动混合心脏亚结构分割：开发与优化。《医学物理与工程科学》，*46*(1)，377–393。

Häntze, H., Xu, L., Mertens, C. J., Dorfner, F. J., Donle, L., Busch, F., Kader, A., Ziegelmayer, S., Bayerl, N., Navab, N., Rueckert, D., Schnabel, J., Aerts, H. J., Truhn, D., Bamberg, F., Weiß, J., Schlett, C. L., Ringhof, S., Niendorf, T., ... Bressem, K. K.（2024）。MRSegmentator：MRI 和计算机断层扫描中的 40 类多模态分割。在 *arXiv [eess.IV]*。arXiv。http://arxiv.org/abs/2405.06463

Herz, C., Fillion-Robin, J.-C., Onken, M., Riesmeier, J., Lasso, A., Pinter, C., Fichtinger, G., Pieper, S., Clunie, D., Kikinis, R., & Fedorov, A. (2017). Dcmqi：一个开源库，可使用 DICOM 标准化传递定量图像分析结果。《癌症研究》，77(21), e87–e90。

Hofmanninger, J., Prayer, F., Pan, J., Röhrich, S., Prosch, H., & Langs, G. (2020). 常规影像中的自动肺分割主要是数据多样性问题，而非方法学问题。《欧洲放射学实验》，4(1), 50。

Hosny, A., Parmar, C., Quackenbush, J., Schwartz, L. H., & Aerts, H. J. W. L. (2018). 放射学中的人工智能。《自然综述：癌症》，18(8), 500–510。

Hugging Face。（无日期）。模型。检索于2026年1月8日，网址：https://huggingface.co/

Huynh, E., Hosny, A., Guthier, C., Bitterman, D. S., Petit, S. F., Haas-Kogan, D. A., Kann, B., Aerts, H. J. W. L., & Mak, R. H.（2020）。放射肿瘤学中的人工智能。《自然综述：临床肿瘤学》，17(12)，771–781。

Isensee, F., Jaeger, P. F., Kohl, S. A. A., Petersen, J., & Maier-Hein, K. H.（2021）。nnU-Net：一种用于基于深度学习的生物医学图像分割的自配置方法。《自然方法》，*18*(2)，203–211。

Jiang, J., & Veeraraghavan, H.（2024）。在真实场景中进行自监督预训练，可赋予医学图像变换器图像采集鲁棒性：应用于肺癌分割。机器学习研究论文集，*250*，708–721。

Kann, B. H., Hosny, A., & Aerts, H. J. W. L. (2021). 人工智能在临床肿瘤学中的应用。《癌细胞》，*39*(7), 916–927。

Koçak, B., Keleş, A., & Köse, F. (2024). 关于人工智能报告指南的元研究：在放射学、核医学和医学影像期刊中，作者和审稿人是否得到了充分鼓励？《诊断与介入放射学》（安卡拉，土耳其）, *30*(5), 291–298。

Korfiatis, P., Kline, T. L., Meyer, H. M., Khalid, S., Leiner, T., Loufek, B. T., Blezek, D., Vidal, D. E., Hartman, R. P., Joppa, L. J., Missert, A. D., Potretzke, T. A., Taubel, J. P., Tjelta, J. A., Callstrom, M. R., & Williamson, E. E. (2025). 在放射学工作流程中实施人工智能算法：挑战与考量。《梅奥诊所学报：数字健康》, *3*(1), 100188。

Kurtzer, G. M., Sochat, V., & Bauer, M. W. (2017). Singularity：用于计算迁移性的科学容器。《公共科学图书馆·综合》，*12*(5)，e0177459。

Lessmann, N., van Ginneken, B., de Jong, P. A., & Išgum, I. (2019). 用于自动椎骨分割与识别的迭代式全卷积神经网络。《医学图像分析》，*53*，142–155。

Li, X., Morgan, P. S., Ashburner, J., Smith, J., & Rorden, C. (2016). 神经影像数据分析的第一步：将 DICOM 转换为 NIfTI。《神经科学方法杂志》，*264*，47–56。

Liao, F., Liang, M., Li, Z., Hu, X., & Song, S. (2019). 使用三维深度泄漏噪声或网络评估肺结节的恶性程度。《*IEEE*神经网络与学习系统汇刊》，*30*(11)，3484–3495。

Liu, X., Faes, L., Kale, A. U., Wagner, S. K., Fu, D. J., Bruynseels, A., Mahendiran, T., Moraes, G., Shamdas, M., Kern, C., Ledsam, J. R., Schmid, M. K., Balaskas, K., Topol, E. J., Bachmann, L. M., Keane, P. A., & Denniston, A. K. (2019). 深度学习在医学影像疾病检测方面的表现与医疗专业人员相比：一项系统综述和荟萃分析。《柳叶刀·数字健康》，*1*(6), e271–e297。

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Jia, Y., Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, ... Xiaoqiang Zheng. (2015年). *TensorFlow*：异构系统上的大规模机器学习。https://www.tensorflow.org/

McGenity, C., Clarke, E. L., Jennings, C., Matthews, G., Cartlidge, C., Freduah-Agyemang, H., Stocken, D. D., & Treanor, D. (2024). 数字病理学中的人工智能：诊断试验准确性的系统综述与荟萃分析。*Npj Digital Medicine*，*7*(1), 114.

Mienye, I. D., Swart, T. G., Obaido, G., Jordan, M., & Ilono, P.（2025）。医学图像分析中的深度卷积神经网络：综述。*Information (Basel)*，*16*(3)，195。

Mildenberger, P., Eichelberg, M., & Martin, E.（2002 年）。DICOM标准简介。《欧洲放射学》，*12*(4)，920–927。Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., Spitzer, E., Raji, I. D., & Gebru, T.（2019）。用于模型报告的模型卡。公平、问责与透明度会议论文集，220–229。Murugesan, G. K., McCrumb, D., Aboian, M., Verma, T., Soni, R., Memon, F., Farahani, K., Pei, L., Wagner, U., Fedorov, A. Y., Clunie, D., Moore, S., & Van Oss, J.（2023年）。用于NCI图像数据共享平台多样化癌症放射学影像集的人工智能生成的标注数据集。见*arXiv [eess.IV]*。arXiv。http://arxiv.org/abs/2310.14897

Najjar, R. (2023年). 重新定义放射学：人工智能在医学影像中的应用综述。《诊断学》（瑞士巴塞尔），*13*(17), 2760。

Niazi, M. K. K., Parwani, A. V., & Gurcan, M. N. (2019). 数字病理学与人工智能。《柳叶刀·肿瘤学》，*20*(5), e253–e261。

Nürnberg, L., Bontempi, D., \{de Ruysscher\}, D., Dekker, A., Hortal, E., Canters, R., & Traverso, A. (2022). 放射治疗计划中心脏亚结构的深度学习分割。《医学物理学：欧洲医学物理学杂志》，*104*(Supplement 1), S76。

Ouyang, W., Beuttenmueller, F., Gómez-de-Mariscal, E., Pape, C., Burke, T., Garcia-López-de-Haro, C., Russell, C., Moya-Sans, L., de-la-Torre-Gutiérrez, C., Schmidt, D., Kutra, D., Novikov, M., Weigert, M., Schmidt, U., Bankhead, P., Jacquemet, G., Sage, D., Henriques, R., Muñoz-Barrutia, A., ... Kreshuk, A. (2022). BioImage 模型动物园：一个由社区推动、提供易用深度学习资源的生物图像分析平台。载于*bioRxiv*。https://doi.org/10.1101/2022.06.07.495102

Pai, S., Bontempi, D., Hadzic, I., Prudente, V., Sokač, M., Chaunzwa, T. L., Bernatz, S., Hosny, A., Mak, R. H., Birkbak, N. J., & Aerts, H. J. W. L. (2024). 癌症影像生物标志物基础模型。《自然·机器智能》，*6*(3), 354–367. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., & Lerer, A. (2017). *PyTorch*中的自动微分。https://openreview.net/pdf?id=BJJsrmfCZ Peng, Y., Kim, J., Feng, D., & Bi, L. (2022). 针对全身多模态PET/CT图像的假阳性减少网络自动肿瘤分割。载于*arXiv [eess.IV]*。arXiv. http://arxiv.org/abs/2209.07705

Pezoa, F., Reutter, J. L., Suarez, F., Ugarte, M., & Vrgoč, D.（2016年4月11日）。JSON Schema 基础。第*25*届万维网国际会议论文集。WWW '16：第25届万维网国际会议，蒙特利尔，魁北克，加拿大。https://doi.org/10.1145/2872427.2883029

Pieper, S., Halle, M., & Kikinis, R. (2005). 3D Slicer. *2004*年 第二届 *IEEE* 生物医学成像国际研讨会：从宏观到纳米 *(IEEE Cat No. 04EX821)*, 632-635 Vol. 1.

Pieper, Steve. (n.d.). *dicomsort*：一个用于自定义排序和重命名 *DICOM* 文件的项目。GitHub。检索于2026年1月8日，网址：https://github.com/pieper/dicomsort Seneviratne, M. G., Shah, N., & Chu, L. (2019). 弥合医疗保健中机器学习的实施鸿沟。*BMJ Innovations*, *6*, 45–47. Sharp, G., Li, R., Wolfgang, J., Chen, G., Peroni, M., Spadea, M., Mori, S., Zhang, J., Shackleford, J., & Kandasamy, N. (05 2010). *PLASTIMATCH——*用于放射治疗图像处理的开源软件套件。Shrestha, A., Watkins, A., Yousefirizi, F., Rahmim, A., & Uribe, C. F. (2024). RT-utils：用于处理放射治疗结构集的轻量级 Python 库。载于*arXiv [physics.med-ph]*。arXiv. http://arxiv.org/abs/2405.06184 Sogancioglu, E., van Ginneken, B., Behrendt, F., Bengs, M., Schlaefer, A., Radu, M., Xu, D., Sheng, K., Scalzo, F., Marcus, E., Papa, S., Teuwen, J., Scholten, E. T., Schalekamp, S., Hendrix, N., Jacobs, C., Hendrix, W., Sanchez, C. I., & Murphy, K. (2024). 胸部X光片上的结节检测与生成：NODE21挑战赛。*IEEE*医学影像学汇刊, *43*(8), 2839–2853. Theriault-Lauzier, P., Cobin, D., Tastet, O., Langlais, E. L., Taji, B., Kang, G., Chong, A.-Y., So, D., Tang, A., Gichoya, J. W., Chandar, S., Déziel, P.-L., Hussin, J. G., Kadoury, S., & Avram, R. (2024). 在医疗护理点将人工智能应用于医学图像和信号的负责任框架：PACS-AI平台。《加拿大心脏病学杂志》, *40*(10), 1828–1840. van Griethuysen, J. J. M., Fedorov, A., Parmar, C., Hosny, A., Aucoin, N., Narayan, V., Beets-Tan, R. G. H., Fillion-Robin, J.-C., Pieper, S., & Aerts, H. J. W. L. (2017). 用于解码放射学表型的计算影像组学系统。《癌症研究》, *77*(21), e104–e107. Vuno. (2022). *Vuno/tiger_*挑战赛。VUNO Inc. https://github.com/vuno/tiger_challenge/

Wasserthal, J., Breit, H.-C., Meyer, M. T., Pradella, M., Hinck, D., Sauter, A. W., Heye, T., Boll, D. T., Cyriac, J., Yang, S., Bach, M., & Segeroth, M.（2023年）。TotalSegmentator：在CT图像中对104种解剖结构进行稳健分割。《放射学：人工智能》，*5*(5), e230024。

Xie, W., Jacobs, C., Charbonnier, J.-P., & van Ginneken, B.（2020）。通过关系建模实现CT 扫描中稳健且高效的肺叶分割。*IEEE*医学影像学汇刊，*39*(8), 2664–2675。Xu, Y., Li, Y., Wang, F., Zhang, Y., & Huang, D.（2025年）。应对基于人工智能的影像组学在癌症影像临床应用中的当前挑战。《医学前沿》，*12*(1674397), 1674397。Youssef, A., Pencina, M., Thakur, A., Zhu, T., Clifton, D., & Shah, N. H.（2023年）。健康领域的人工智能模型外部验证应由反复进行的本地验证取代。《自然·医学》，*29*(11), 2686–2687。Ziegler, E., Urban, T., Brown, D., Petts, J., Pieper, S. D., Lewis, R., Hafey, C., & Harris, G. J.（2020）。开放健康影像基金会查看器：一个可扩展的开源框架，用于构建支持癌症研究的基于网络的影像应用程序。《*JCO*临床癌症信息学》，*4*(4), 336–345。