Clinical deployment and validation of a radiology artificial intelligence system for COVID-19

Behrooz Hashemian (bhashemian@mgh.harvard.edu)
Mass General Brigham, Boston, MA

Aman Manchanda
Mass General Brigham, Boston, MA

Matthew D. Li
Department of Radiology, Massachusetts General Hospital, Boston, MA

Parisa Farzam
Department of Neurosurgery, Brigham and Women's Hospital, Boston, MA

Suma D. Dash
Mass General Brigham, Boston, MA

Alvin Ihsan
NVIDIA, Santa Clara, CA

Jiahui Guan
NVIDIA, Santa Clara, CA

Risto Haukioja
NVIDIA, Santa Clara, CA

Nishanth Thumbavanam Arun
Athinoula A. Martinos Center for Biomedical Imaging, Charlestown, MA

Ram Naidu
Mass General Brigham, Boston, MA

Thomas Schultz
Mass General Brigham, Boston, MA

Katherine P. Andriole
MGH & BWH Center for Clinical Data Science, Boston, MA

Jayashree Kalpathy-Cramer
MGH & BWH Center for Clinical Data Science, Boston, MA

Keith J. Dreyer
Mass General Brigham, Boston, MA

Research Article

Keywords: Artificial Intelligence, Medical Imaging, Clinical Inference System, COVID-19, Chest Radiograph

Posted Date: August 19th, 2020

DOI: https://doi.org/10.21203/rs.3.rs-61220/v1

Clinical deployment and validation of a radiology artificial intelligence system for COVID-19

Behrooz Hashemian[1,2,3,*], Aman Manchanda[1,2], Matthew D. Li[3], Parisa Farzam[4], Suma D. Dash[1,2], Alvin Ihsani[7], Jiahui Guan[7], Risto Haukioja[7], Nishanth Thumbavanam Arun[6], Ram Naidu[1,2], Thomas Schultz[1], Katherine P. Andriole[2,5], Jayashree Kalpathy-Cramer[2,3,6], and Keith J. Dreyer[1,2,3]

[1]Mass General Brigham, Boston, MA
[2]MGH & BWH Center for Clinical Data Science, Boston, MA
[3]Department of Radiology, Massachusetts General Hospital, Boston, MA
[4]Department of Neurosurgery, Brigham and Women's Hospital, Boston, MA
[5]Department of Radiology, Brigham and Women's Hospital, Boston, MA
[6]Athinoula A. Martinos Center for Biomedical Imaging, Charlestown, MA
[7]NVIDIA, Santa Clara, CA
[*]bhashemian@mgh.harvard.edu

ABSTRACT

The global COVID-19 pandemic has disrupted patient care delivery in healthcare systems world-wide. For healthcare providers to better allocate their resources and improve the care for patients with severe disease, it is valuable to be able to identify those patients with COVID-19 who are at higher risk for clinical complications. This may help to optimize clinical workflow and more efficiently allocate scarce medical resources. To this end, medical imaging shows great potential and artificial intelligence (AI) algorithms have been developed to assist in diagnosing and risk stratifying COVID-19 patients. However, despite the rapid development of numerous AI models, these models cannot be clinically useful unless they can be deployed in real-world environments in real-time on clinical data. Here, we propose an end-to-end AI hospital-deployment architecture for COVID-19 medical imaging algorithms in hospitals. We have successfully implemented this system at our institution and it has been used in prospective clinical validation of a deep learning algorithm potentially useful for triaging of patients with COVID-19. We demonstrate that many orchestration processes are required before AI inference can be performed on a radiology studies in real-time with the AI model being just one of the components that make up the AI deployment system. We also highlight that failure of any one of these processes can adversely affect the model's performance.

Introduction

Severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) and its resultant disease COVID-19 has led to a global pandemic of COVID-19[1]. This has overwhelmed healthcare systems and posed new global challenges for patient care management[2–4]. Hospitals have reported many challenges in providing care to patients, from difficulty in maintaining adequate staffing to shortage of medical resources such as ICU beds and ventilators[5]. This has created demand for protocols for better resource management and optimization of hospital workflows. Rapidly diagnosing and identifying the severity risk of COVID-19 for each patients is valuable in guiding their care while optimizing hospital resource utilization[6,7].

Real-time reverse transcription–polymerase chain reaction (rRT-PCR) has been widely used for detecting, tracking, and studying COVID-19 across healthcare systems; however, the concerns about availability and stability of these tests[8] along with their moderate sensitivity in diagnosis[9–11] has motivated clinicians to seek complimentary tools for diagnosis, prognosis and treatment of this disease. To this end, medical imaging has been used to increase diagnostic accuracy[12,13], evaluate disease severity[14,15], and follow up on infected patients[16].

The morphological changes in the peripheral lungs of patients with moderate or severe disease[17–20] create imaging markers in chest radiographs. Chest imaging can play an important role in the clinical management of patients with COVID-19 that have clinical manifestations or worsening respiratory symptoms due to COVID-19[21,22]. Though the American College of Radiology recommends against the use of chest CT alone for COVID-19 diagnosis or screening[23], front-line clinicians order chest radiographs or computed tomography (CT) studies to help guide their clinical decisions. Radiologists interpret these studies and render diagnoses, assess disease severity, and describe change compared to prior imaging studies. To mitigate the burden on radiologists, while providing the highest quality care for patients, there has been tremendous effort to develop

novel image processing approaches using machine learning algorithms[24], particularly for COVID-19 diagnosis and prognosis[25]. These artificial intelligence (AI) models exploit and build upon medical imaging modalities such as chest CT scans[26–32], chest radiographs[33–40], and lung ultrasound[41]

However, for any of these AI models to be useful in assisting clinicians in the care of COVID-19 patients, they require a robust and reliable AI deployment system[42]. Deployment is often a difficult step because clinical radiology infrastructure is not designed for easily embedding third-party systems, and doing so while maintaining context sensitivity and seamlessly embedding such systems into the radiologist workflow requires knowledge of hospital information system integration standards and often product-specific knowledge. This not only affects evaluation of these models since they are usually not tested on prospective real-world data but also diminishes their utility since they cannot be integrated into the radiology workflow.

In this work, we detail an AI deployment system that leverages existing radiology infrastructure of hospitals and we combine it with the state-of-the-art technology in machine learning systems. We deploy an AI model for automated assessment of pulmonary disease severity in chest radiographs of patients with suspected or confirmed COVID-19[43]. We illustrate the technical challenges with end-to-end deployment of this AI algorithm during the COVID-19 pandemic, from accessing the radiology Digital Imaging and Communication in Medicine (DICOM)[1] live-feed to storing AI output in a SQL database. Training and testing an AI model is only the first step towards making such a model accessible and potentially useful in clinical workflows. Where possible, we attempt to generalize our findings as they likely apply to others trying to deploy AI models that operate on medical images.

Methods

We designed, developed, and deployed an AI deployment system, based upon radiology infrastructures, that receives real-time clinical studies from our institution, and ran our AI model to perform inference on chest radiographs of patients with suspected or confirmed COVID-19. Research in developing this AI model was exempted by our hospital's Institutional Review Board, with waiver of informed consent.

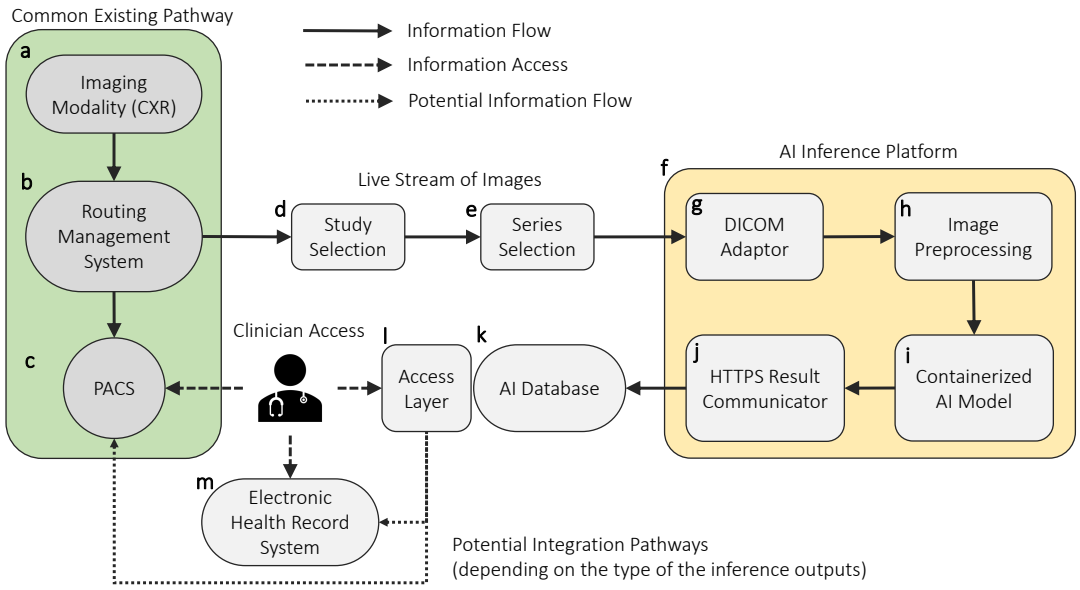

Figure 1. Workflow for clinical deployment of COVID-19 AI model

Deep learning model development

We implement a deep learning model that assesses pulmonary consolidative disease severity on chest radiographs of COVID-19 patients[43]. The algorithm incorporates a convolutional Siamese neural network-based approach[44] to provide a measure of

[1] https://www.dicomstandard.org

disease severity[14]. This approach can help clinicians assess risk for worsening clinical outcomes and can also be used to track change over time from one radiograph to another. To evaluate the AI algorithm on prospective clinical studies, we embed the algorithm into our AI deployment system (Figure 1) and perform inference on imaging data acquired in real-time.

Real-time routing and selection of medical images

The medical imaging examinations assessed by an AI model during inference must have similar characteristics to the training data. Example characteristics include acquisition modality, image view, patient position, pixel spacing, intensity distribution, reconstruction kernel, etc. We categorize the process of providing the most appropriate medical image to a specific AI model into three related tasks: study selection, series selection, and image pre-processing. To automate the ingestion and distribution of medical imaging studies, we employ Compass[2], which is a versatile routing workflow manager. In the following sections, we explain the methods used for study and series selection.

Study Selection

Our COVID-19 AI model requires a single anteroposterior (AP) chest radiograph image (typical of portable bedside imaging) in DICOM image format as input. Therefore we first need to effectively and automatically identify the studies of interest for a particular AI model, and forward them to the AI inference platform. This task is performed by a routing management system (Figure 1-b) that receives images acquired from all of the imaging modalities at our hospital (Figure 1-a) and selects the specific imaging examinations using DICOM data elements. Due to a lack of standardized labeling of images, we devised a rule-based algorithm that uses a combination of standard and hospital-specific private data elements to identify the relevant studies (Figure 1-d). In our case, the most effective element was a private DICOM tag named Exam Code, that helped us isolate all chest radiographs. This DICOM tag was originally developed for billing purposes and is specific to our hospital.

Series Selection

After identifying a study of interest, our solution further processes the data to identify the series of interest since a study may consist of multiple series with post-processed or marked up images. For chest radiographs, each series contains a single image, so selecting the correct series results in selection of the correct image. This may not be the case for other modalities (e.g. CT or MRI studies), and thus in-depth domain knowledge of the modality, and attributes of the input images for machine learning model is often required. For our solution, we select the series of interest by filtering them based on DICOM data elements, series acquisition time, and our knowledge of hospital workflow. At our hospital, the `Series Description` DICOM tag (0008,103E) is standardized to account for different views so we rely on this tag and include any series containing `AP` in the `Series Description`. We also use the `Derivation Description` DICOM tag (0008,2111) to remove series with *CATH* labels, which reflect chest radiographs post-processed to accentuate lines and tubes, but would not be appropriate as input for our model (Figure 1-e). The resulting series are sent to our live AI inference platform (Figure 1-f).

Live AI inference platform

At the core of our AI deployment system is our AI inference platform (Figure 1-f), which is capable of efficiently running the developed AI model on the presented images. Our platform uses NVIDIA's Clara Deploy Software Development Kit (SDK)[3] running on an NVIDIA DGX-1[4] server. The Clara Deploy SDK provides a container-based deployment framework for multi-AI workflows. It uses Kubernetes[5] for container orchestration, and enables defining multi-staged inference pipelines as described below. Figure 2 depicts the System Architecture for the Clara Deploy SDK.

Clara Pipeline

A Clara pipeline is a collection of containers that are configured to work together to execute a medical image processing task. Listing 2 shows the Clara pipeline definition for our model. We use a Docker[6] image tagged "pxs-score-app" for pre-processing the image and for executing inference, and another Docker container tagged "results-communicator-app" for posting inference results to our results database.

Clara DICOM Adapter

The Clara DICOM Adapter provides the ability to receive DICOM studies as a DICOM Service Class Provider (SCP), and for each study, schedule a job which executes an instance of the pipeline. We assign an Application Entity (AE) Title of 'PxsScore' to the DICOM Adapter, and associate the AE Title with the Clara Pipeline. The hospital's routing management system forwards matching studies to the DICOM Adapter using the 'PxsScore' AE Title which in turn triggers our pipeline. Listing 1 shows a sample DICOM Adapter configuration.

[2] http://www.laurelbridge.com/products/compass
[3] https://docs.nvidia.com/clara/deploy
[4] https://www.nvidia.com/en-us/data-center/dgx-1
[5] https://kubernetes.io
[6] https://www.docker.com/

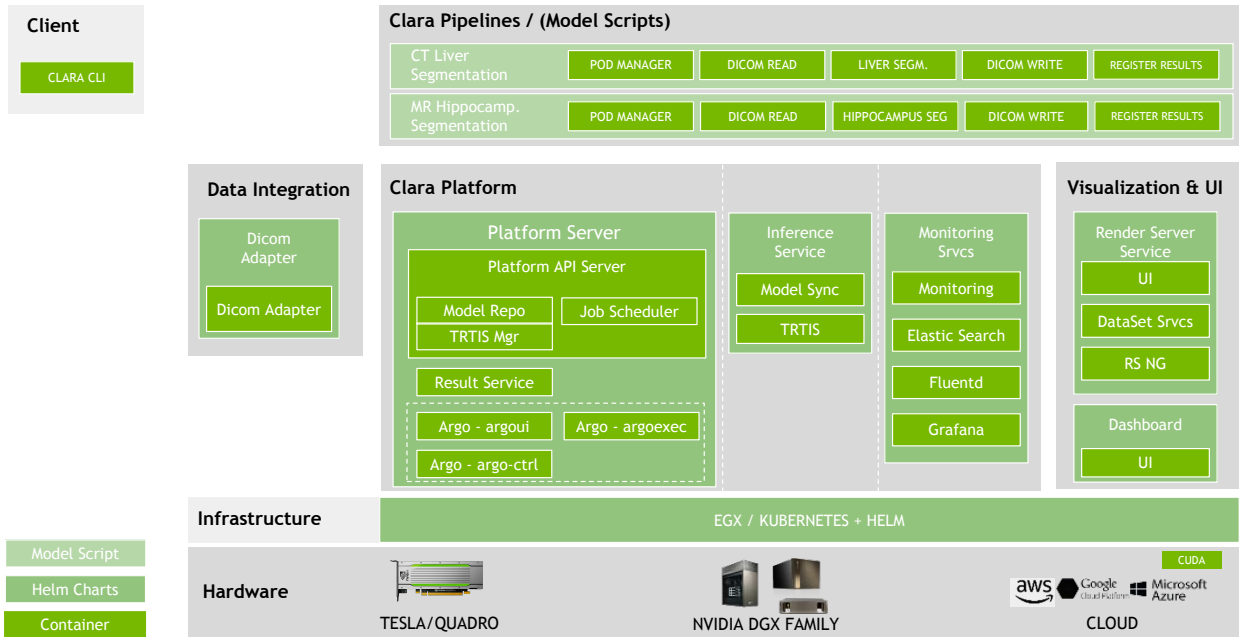

Figure 2. Clara system architecture. The Clara Deploy SDK is run via a collection of components that are deployed via model scripts, helm charts, and containers. The image is courtesy of NVIDIA.

Image Pre-processing

Image pre-processing (Figure 1-h) occurs in a Docker container that includes all scripts needed to prepare the medical images, such as reading the DICOM images, and converting them into tensors, before loading and running the model described. When working with containerized AI models, prepossessing code is often bundled in the same container as the AI model (Figure 1-i) and this is how our system is configured. However, when using inference serving software like NVIDIA Triton Inference Server[7] it can be more efficient to separate the pre-processing module from the AI model module.

Containerized AI Model

Our model development workflow produces a Docker image[8] that contains the pre-processing code described, the model, and the code to post-process and store the results. The Docker image is configured to read DICOM files[9] from an input folder, and write results to an output folder. Clara Deploy pulls this image and creates a container for each study received, and supplies the study in the container's input folder. Once the AI model container has completed execution, Clara Deploy takes the output and provides it as the input for the next container in the line, the HTTPS Result Communicator.

AI Result Management

We developed a single JSON-based result model that summarizes all findings including references to any binary files produced. We expect binary output files by AI models to be in a standard DICOM format, including DICOM Segmentation, Secondary Capture Image, and Grayscale Softcopy Presentation States (GSPS). Listing 3 shows this JSON model and the fields `createdAt` and `updatedAt` indicate when inference is started and when it is completed respectively. Multiple inference results may be output as a list of `results` objects. For each result, `results.code` field specifies a standard format or lexicon such as Common Data Elements (CDE)[10], RadLex[11] or SNOMED[12] to describe the findings. Each result includes one `results.derivedFrom` object that lists the series and SOP instance UIDs that were selected as the input to the inference algorithm. It may also include a `results.presentOn` object that contains information on how and where a clinical viewer may display the results. Finally, the `results.refEntities` field references items in the `entities` object that lists files and other binary output from the model.

[7] https://docs.nvidia.com/deeplearning/triton-inference-server
[8] https://docs.docker.com/engine/reference/commandline/image
[9] http://dicom.nema.org/medical/dicom/current/output/html/part10.html
[10] https://www.radelement.org/home/sets
[11] http://www.radlex.org
[12] https://www.snomed.org

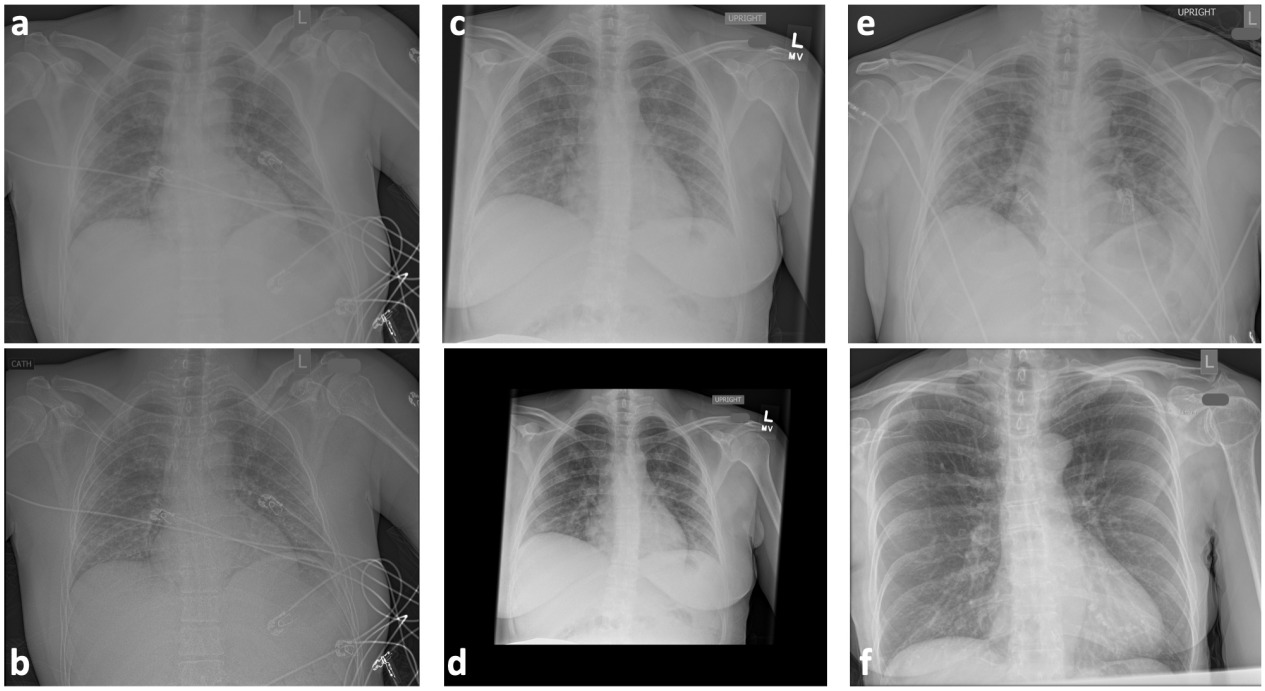

Figure 3. Examples of DICOM chest radiograph images. The top row includes acceptable model input images and on the bottom are their corresponding undesirable images from the same study. (b) is an example of CATH image (post-processed image that accentuates lines and tubes), (d) is an example of screen-shot image, and (f) is an example image from a wrong patient included with the correct image above.

The HTTP results communicator (Figure 1-j) takes the JSON-based outputs along with the associated binary files and sends them to a database service using a RESTful interface. We use a SQL Server database to store the results in a JSON format along with several indices. e.g. Inference ID, PatientID, and StudyInstanceUID (Figure 1-k). We leverage SQL Server's native capability to query results directly from JSON (and not from a relational structured database) to create customized views for each type of users, such as radiologists, researchers, and technical support teams (Figure 1-l).

Monitoring live inference

Monitoring the health of the live inference system is critical. We use the Clara Management Console[13], the Clara Deploy CLI[14], and the kubectl[15] tool to monitor all jobs initiated by the DICOM adapter. The console provides a user interface that lists new jobs as they are created, their current status, and upon inference completion, it lists the job duration. The Clara Deploy SDK also stores logs from containers in Elasticsearch[16] and uses a Grafana[17] frontend which we use to view logs and debug issues.

Configurations and APIs

In Listing 1 and Listing 2, we provide examples of Clara Deploy configuration files for DICOM Adaptor and the Clara pipeline that we use in this clinical deployment. Listing 3 depicts the result JSON API that we designed to standardize and manage the AI model outputs.

Listing 1. Clara Deploy DICOM Adapter configuration. The Clara Deploy DICOM Adapter receives DICOM studies and triggers the pipeline containing the COVID-19 AI model.

```
dicom:
  scp:
    port: 104
    ae-titles:
      - ae-title: PxsScore
```

[13] https://docs.nvidia.com/clara/deploy/sdk/clara-ux/public/docs/README.html
[14] https://docs.nvidia.com/clara/deploy/ClaraCLI/index.html
[15] https://kubernetes.io/docs/reference/kubectl/overview
[16] https://www.elastic.co/elasticsearch
[17] https://www.grafana.com

```
    max-associations: 4
    verification:
      enabled: true
      transfer-syntaxes:
        - "1.2.840.10008.1.2" #Implicit VR Little Endian
        - "1.2.840.10008.1.2.1" #Explicit VR Little Endian
        - "1.2.840.10008.1.2.2" #Explicit VR Big Endian
    log-dimse-datasets: false
    reject-unknown-sources: true
    sources:
      - host-ip: 192.168.1.2
        ae-title: RMS
pipeline-mappings:
  - name: pxs-score
    clara-ae-title: PxsScore
    pipeline-id: 112233445566778899aabbccddeeff00
```

Listing 2. Clara Deploy Pipeline definition. In this example, the pipeline contains two Docker containers: one for the AI model and one for the HTTPS result communicator.

```
api-version: 0.4.0
name: pxs-score-pipeline-0-1
operators:
- name: pxs-score
  description: PXS Score on Chest XRay
  container:
    image: pxs-score-app
    tag: 0.1
  input:
  - path: /PXS_score/dicoms
  output:
  - path: /PXS_score/outputs
    name: results
- name: result-communicator-0-1
  description: Reads JSON results, and sends it to the database.
  container:
    image: result-communicator-app
    tag: 0.1
  input:
  - from: pxs-score
    name: results
    path: /input
```

Listing 3. Inference results JSON file. This structured file is used to communicate inference results on a given study.

```
{
  "version": "0.2.4",
  "id": "",
  "status": "SUCCEEDED",
  "studyInstanceUid": "1.2.345.678.9999",
  "patientId": "MRN457",
  "accessionNumber": "E12346",
  "siteId": "Hospital",
  "createdAt": "2020-05-23T16:57:44-05:00",
  "updatedAt": "2020-05-23T16:58:03-05:00",
  "exitCode": 0,
  "errorMessage": "",
  "model": {
    "id": "PxsScore~0.1",
    "name": "PxsScore",
    "displayName": "PxsScore",
    "version": "0.1",
    "author": "Data Science",
    "description": "PXS score model on chest x-rays",
    "createdAt": "2020-02-29T17:52:40.844392531Z"
  },
  "results": [
    {
      "code": "partners.clara.pxs.score",
```

```
      "value": "2.45",
      "description": "PXS Score",
      "derivedFrom": [
        {
          "refSeriesInstanceUid": "1.2.840.113.111111",
          "refSopInstanceUid": [
            "2.4.5.61.2.826.0.1.3680043.9.99999.222222",
            "2.4.5.61.2.826.0.1.3680043.9.99999.222223"
          ]
        },
        {
          "refSeriesInstanceUid": "1.2.840.113.333333",
          "refSopInstanceUid": [
            "2.4.5.61.2.826.0.1.3680043.9.99999.444444",
            "2.4.5.61.2.826.0.1.3680043.9.99999.444445"
          ]
        }
      ],
      "presentOn": [
        {
          "refSeriesInstanceUid": "1.2.840.113.111111",
          "refSopInstanceUid": [
            "2.4.5.61.2.826.0.1.3680043.9.99999.222222",
            "2.4.5.61.2.826.0.1.3680043.9.99999.222223"
          ]
        },
        {
          "refSeriesInstanceUid": "1.2.840.113.333333",
          "refSopInstanceUid": [
            "2.4.5.61.2.826.0.1.3680043.9.99999.444444",
            "2.4.5.61.2.826.0.1.3680043.9.99999.444445"
          ]
        }
      ],
      "refEntities": [
        "IM-0002-0001-0001.dcm"
      ]
    }
  ],
  "entities": [
    {
      "name": "IM-0001-0001.dcm",
      "uri": "",
      "type": "dicom"
    }
  ]
}
```


Results and Discussion

Clinical deployment of medical imaging algorithms in hospitals has several components, and the AI model itself is just one of them. Developing a robust, reliable AI model requires real-world validation beyond a controlled environment that is typical during the model training phase. There are many orchestration processes required before AI inference can be performed on radiology images in real-time and failure of any one of these processes can affect model performance. To ensure successful clinical deployment of AI models, the entire pipeline from data acquisition (Scanners, PACS) to model results visualization must be considered.

Though our solution uses specific software components, the proposed architecture depicted in Figure 1 is implementation agnostic and portable to a variety of off-the-shelf or custom software solutions available in clinics and hospitals. It can be used as a guideline for an agile and effective deployment of AI models based upon existing radiology infrastructure to address time-sensitive issues like COVID-19.

Appropriate image selection

Routing correct input imaging examinations to the AI model is a critical step in clinically translating live inference systems for medical imaging. Unfortunately this task is frequently overlooked during model development even though it can severely affect model performance when deployed into live clinical workflow. The solution must consider multiple variables in study selection

such as the source equipment, clinician workflow, hospital policy and procedures, and the software and infrastructure that routes the images to the inference system. Chest radiographs at our institution may have more than one image per study, often a combination of a standard image with a *CATH* image indicating a post-processed image copy (Figure 3-b) or a *screenshot* which may contain some radiologist or technologist mark-up (Figure 3-d). There are a small portion of studies with more than three AP images (about 2%). These are often caused by the need to rescan based upon poor image quality due to non ideal patient positioning, which may be related to body habitus or critical illness. We flagged those studies for manual review.

In our initial attempt at series selection, we filtered the series purely based on their naming, which resulted in approximately 50% of cases being missed. Using the approach described in Methods, we increased our series selection yield to greater than 95%. The small proportion of missed image views are due to mislabeling by the technologist at acquisition time. Figure 3-f shows one of these examples. While prospectively collecting clinical data for performed studies, such erroneously labeled studies are identified by the presence of more than one inference result or exception that the model may raise during processing of these images. These are manually reviewed and feedback is sent to the relevant information technology teams to correct this mistake in the Picture Archiving and Communication System (PACS).

Challenges with image selection

We show that relying exclusively on the series name is not a reliable approach in determining which images should be used as real-time input to the model and present a solution to select the appropriate image. Individual healthcare sites often develop internal standards for identifying DICOM images, which may not be generalizable across hospital systems. Our solution relies on the `Series Description` and `Derivation Description` DICOM data elements, which are shown to be quite reliable at our hospital. Note, however, that these data elements are optional and may be left blank or contain site- or vendor-specific information. Thus there is a risk that relying on these data elements alone may result in unknown consequences such as, but not limited to: omission of a relevant series, inclusion of an irrelevant series, or even a total failure of the series selection algorithm. As a more effective alternative to our highlighted approach, we are working on developing separate AI models for selecting specific images which will be trained on DICOM metadata as well as on the pixel images. This approach can not only generalize across healthcare systems but can also increase the accuracy of image selection for each site.

However, manual checking is required in the case in which a technologist mistakenly captures an image in a wrong study (Figure 3-f). To mitigate this issue, our solution flags studies on which inference has been run more than once, so that a clinician may review and correct results as needed.

Prospective clinical validation of a COVID-19 AI model

The PXS Score AI model, described in Methods, can potentially aid radiologists in making consistent disease severity assessments in their report recommendations and aid front-line clinicians in quantifying COVID-19 patient risk assessments to guide patient management. It can be considered as an additional data point that correlates with clinical outcomes (higher numeric PXS score are associated with subsequent intubation or death)[43]. Although a detailed assessment of this AI model is beyond the scope of this manuscript, here we show how our AI deployment system enables physicians and scientists to evaluate their model performance in real time and on prospective medical examinations. In 135 patients admitted to our hospital from 4/24/2020 to 4/29/2020 with RT-PCR confirmed COVID-19 and an admission chest radiograph, PXS scores were calculated in real-time using our AI deployment system. Of these patients, 22 met a composite outcome of ICU admission/transfer, intubation, and/or death within 3 days of hospital admission. The PXS score is predictive of this composite outcome, with an area under the receiving operator characteristic curve of 0.81 (bootstrap 95 percent confidence interval 0.70-092) (Figure 4). This finding confirms retrospective results found in PXS score model development[43].

Traceability of the inference results

To aid in continuous improvement of the model, our solution recorded several items in the output JSON format described in Methods. For each result stored in the database, we also store information to identify the model, the patient, as well as the Study Instance UID, Series Instance UID and SOP Instance UID used to obtain results. We also stored the date and time of inference, exit code, and any error messages output by the model. This allowed us to trace what happened to the study from the originating modality, all the way to when the inference results were generated.

System reliability

For an AI algorithm to be deployed in a clinical setting, robust and reliable integration of the algorithm is crucial. Our solution include a reliable system with monitoring mechanisms in place (see Methods for further details). However our existing AI deployment system risks failure to perform or record inference in a variety of scenarios such as application errors, file system errors, network errors, etc. We intend to mitigate such errors and improve reliability in future versions. An exercise for the future is to ensure that the Clara Deploy SDK provides at-least-once guarantee for all studies it receives - that is to say that it guarantees running on every study at least once. On rare occasions where there is a system error, it may run inference more

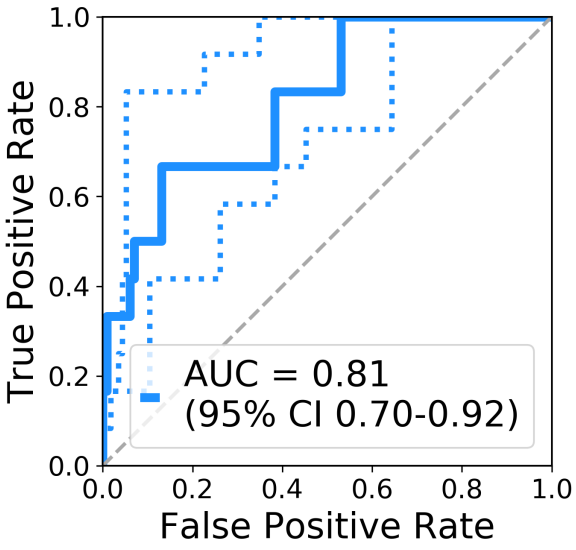

Figure 4. Receiver operating characteristic curve for prospective validation of PXS score-based (AI model) prediction of composite outcome of ICU admission/transfer, intubation, and/or death within 3 days of hospital admission for COVID-19 patients at one hospital. The PXS score (AI model output) is calculated in real-time using our AI deployment system. AUC = area under the curve; CI = confidence interval.

than once, but at no time shall it fail to attempt inference on a study. We are further investigating aggregating logs and metrics from all subsystems to a common service, such as Elasticsearch, for more efficient analysis.

Performance and Speed

Using our AI deployment system, we prospectively validate an AI model for COVID-19 outcome prediction in real-time with a typical pipeline execution time of approximately 20 seconds per study. The average AI inference time, the time from when the input images are written into the container to when the results are created, is 12.9 seconds with standard deviation of 3.5 seconds. We have tested this system for more than one month and with approximately 10,000 prospective studies at one hospital. This is acceptable performance of our system for our chest radiograph use-case; however, this may become unacceptably slower when deploying models on large 3D medical studies. The inference time, in these situation, can surpass tens of minutes unless using an optimized containerized AI model. As mentioned in Methods, one can take advantage of NVIDIA Triton Inference Server to efficiently run models concurrently on multiple GPUs maximizing utilization. We further intend to review the flow of studies to and through the AI deployment system, to ensure that we minimize the overall time before results are committed to the database.

Scalability and extensibility

At present, our solution is constrained to a single Kubernetes Node, with another node currently in development. We are able to scale our solution by configuring the Routing Management System to direct studies to different Clara Deploy installations, and balance the deployments based on the load of studies coming from the hospital and the resources required to execute inference. However, we realize that this solution does not make efficient use of system resources. Different models and workflows have different requirements for when and how inference results are made available. Where possible, we will design future elements of our solution as single-function microservices that can be scaled horizontally across multiple Kubernetes Nodes as needed. We intend to ensure that the infrastructure and software microservices are designed to take into account larger scale deployment with many models and optimizing for metrics such as fast run-time performance, flexibility to scale horizontally, adequate storage space, computational power etc.

We have designed our work to be extensible to various imaging and non-imaging AI models. Our results management API, described in Methods, is designed to encompass variability in the type of model outputs, from a single score and textual output to image segmentation and image overlays to a combination of medical diagnoses. The JSON output format allows us flexibly

to add multiple findings and reference them to the corresponding input source.

Security

Security and hardening must be considered for the AI inference platform including but not limited to the Operating System, system tools and utilities, Kubernetes, Docker, Clara Deploy SDK, and all of the custom operators and services. At present, model developers provide Docker images that are accepted into the system as a black box. Administrative access to the platform is restricted, and we rely on the inherent security in running models in a private, isolated environment.

We plan to implement further restrictions such as limiting network access, CPU, GPU, and disk resources, and the maximum amount of time allowed for a model to run. We are further researching the development of a base Docker image that model developers could then build on top of that would allow us to better maintain and secure the model code - this is a controversial topic as we have found model developers do not like such restrictions being placed on them.

Best practices in software development and deployment, such as the use of peer reviews, source control, and continuous integration, continuous delivery, and continuous deployment workflows (CI/CD), must be utilized for model development, infrastructure development, and microservice development alike. Further consideration must be given for maintenance so the overall system continues to receive necessary software updates and security patches for open source and proprietary components in a timely manner.

Monitoring and alerting

In this work, we monitor various operators and services including image routed to the AI inference platform, declared pipelines, created jobs, communicated payloads, and individual Docker container's logs through different command-line and graphical user interfaces. At present, we do not have an automated software solution in place to collect and gather such metrics in a central dashboard. While Clara Deploy does allow us to collect metrics on system resources such as CPU, GPU, disk usage, we intend to develop a metrics collection solution that allows us to measure resources consumed at various levels in the system including for each model, each pipeline defined in Clara Deploy, various exam codes received by the AI inference platform, each Clara Deploy operator, Clara Deploy internals, etc.

Integration with electronic medical record system

The output of our medical imaging AI model is stored in a SQL database; however, its use in clinical practice to inform decision-making will depend on nascent and ongoing prospective clinical validation studies. Determining the optimal mode of delivery of this information to the end-user is outside of the scope of this article, but eventual integration with the radiology information system and/or electronic medical record will be critical. But there is a wide spectrum of problems between the AI model producing an output and the end-user receiving that output that must be considered and dealt with, to allow for the successful clinical implementation of an AI model.

Conclusion

The reliable development of COVID-19 AI models and timely transition of these AI models into clinically validated systems that can effectively improve the patient care require a carefully built and integrated AI deployment system. The complexity of such systems necessitates orchestration of many components from image acquisition at the scanner to inference results presentation at the point-of-care. The AI deployment system is also a key contributor in further developing and validating AI models by exposing the models to prospective patient data and allowing developers and clinicians to review model results and validate them outside the controlled environment in which AI models are typically trained. We describe an architectural design and clinical implementation of such a system in our hospital. We show how this agile technology enables physicians and researchers at our institution to validate their AI model to determine the severity of patients with COVID-19 for providing a better patient care. We also highlight the challenges we encountered and how we solved them. For successful clinical deployment of AI models, it is important to consider the entire pipeline from data acquisition (medical imaging scanners) to model inference to results management and presentation.

References

1. World Health Organization. WHO Director-General's opening remarks at the media briefing on COVID-19 - 11 March 2020. https://www.who.int/dg/speeches/detail/who-director-general-s-opening-remarks-at-the-media-briefing-on-covid-19—11-march-2020 (11 March 2020).
2. Moghadas, S. M. *et al.* Projecting hospital utilization during the covid-19 outbreaks in the united states. *Proc. Natl. Acad. Sci.* **117**, 9122–9126 (2020).
3. Willan, J., King, A. J., Jeffery, K. & Bienz, N. Challenges for nhs hospitals during covid-19 epidemic. *BMJ* **368** (2020).

4. Armocida, B., Formenti, B., Ussai, S., Palestra, F. & Missoni, E. The italian health system and the covid-19 challenge. *The Lancet Public Heal.* **5**, e253 (2020).

5. Grimm, C. A. Hospital experiences responding to the covid-19 pandemic: Results of a national pulse survey march 23–27, 2020. Tech. Rep., Office of Inspector General, U.S. Department of Health and Human Services (2020).

6. Judson, T. J. *et al.* Rapid design and implementation of an integrated patient self-triage and self-scheduling tool for COVID-19. *J. Am. Med. Informatics Assoc.* **27**, 860–866 (2020).

7. Reeves, J. J. *et al.* Rapid response to COVID-19: health informatics support for outbreak management in an academic health system. *J. Am. Med. Informatics Assoc.* **27**, 853–859 (2020).

8. Li, Y. *et al.* Stability issues of rt-pcr testing of sars-cov-2 for hospitalized patients clinically diagnosed with covid-19. *J. Med. Virol.* (2020).

9. Wang, W. *et al.* Detection of SARS-CoV-2 in Different Types of Clinical Specimens. *JAMA* **323**, 1843–1844 (2020).

10. Tahamtan, A. & Ardebili, A. Real-time rt-pcr in covid-19 detection: issues affecting the results. *Expert. Rev. Mol. Diagn.* **20**, 453–454 (2020).

11. Xie, X. *et al.* Chest ct for typical 2019-ncov pneumonia: Relationship to negative rt-pcr testing. *Radiology* 200343 (2020).

12. Fang, Y. *et al.* Sensitivity of Chest CT for COVID-19: Comparison to RT-PCR. *Radiology* 200432 (2020).

13. Ai, T. *et al.* Correlation of Chest CT and RT-PCR Testing in Coronavirus Disease 2019 (COVID-19) in China: A Report of 1014 Cases. *Radiology* 200642 (2020).

14. Li, M. D. *et al.* Siamese neural networks for continuous disease severity evaluation and change detection in medical imaging. *npj Digit. Medicine* **3**, 48 (2020).

15. Hosseiny, M., Kooraki, S., Gholamrezanezhad, A., Reddy, S. & Myers, L. Radiology perspective of coronavirus disease 2019 (covid-19): lessons from severe acute respiratory syndrome and middle east respiratory syndrome. *Am. J. Roentgenol.* **214**, 1078–1082 (2020).

16. Pan, F. *et al.* Time course of lung changes at chest ct during recovery from coronavirus disease 2019 (covid-19). *Radiology* **295**, 715–721 (2020).

17. Guan, W.-j. *et al.* Clinical characteristics of coronavirus disease 2019 in china. *New Engl. J. Medicine* **382**, 1708–1720 (2020).

18. Ackermann, M. *et al.* Pulmonary vascular endothelialitis, thrombosis, and angiogenesis in covid-19. *New Engl. J. Medicine* (2020).

19. Hariri, L. & Hardin, C. C. Covid-19, angiogenesis, and ards endotypes. *New Engl. J. Medicine* (2020).

20. Liu, X. *et al.* Temporal radiographic changes in COVID-19 patients: relationship to disease severity and viral clearance. *Sci. Reports* **10**, 10263 (2020).

21. Rubin, G. D. *et al.* The role of chest imaging in patient management during the covid-19 pandemic: A multinational consensus statement from the fleischner society. *Radiology* 201365 (2020).

22. Zhao, W., Zhong, Z., Xie, X., Yu, Q. & Liu, J. Relation between chest ct findings and clinical conditions of coronavirus disease (covid-19) pneumonia: a multicenter study. *Am. J. Roentgenol.* **214**, 1072–1077 (2020).

23. American College of Radiology. ACR Recommendations for the use of Chest Radiography and Computed Tomography (CT) for Suspected COVID-19 Infection. https://www.acr.org/Advocacy-and-Economics/ACR-Position-Statements/Recommendations-for-Chest-Radiography-and-CT-for-Suspected-COVID19-Infection (22 March 2020).

24. Litjens, G. *et al.* A survey on deep learning in medical image analysis. *Med. Image Analysis* **42**, 60 – 88 (2017).

25. Wynants, L. *et al.* Prediction models for diagnosis and prognosis of covid-19: systematic review and critical appraisal. *BMJ* **369** (2020).

26. Wang, J. *et al.* Prior-attention residual learning for more discriminative covid-19 screening in ct images. *IEEE Transactions on Med. Imaging* (2020).

27. Fan, D. *et al.* Inf-net: Automatic covid-19 lung infection segmentation from ct images. *IEEE Transactions on Med. Imaging* (2020).

28. Han, Z. *et al.* Accurate screening of covid-19 using attention based deep 3d multiple instance learning. *IEEE Transactions on Med. Imaging* (2020).

29. Zhang, K. *et al.* Clinically applicable ai system for accurate diagnosis, quantitative measurements, and prognosis of covid-19 pneumonia using computed tomography. *Cell* (2020).

30. Wang, X. *et al.* A weakly-supervised framework for covid-19 classification and lesion localization from chest ct. *IEEE Transactions on Med. Imaging* (2020).

31. Ouyang, X. *et al.* Dual-sampling attention network for diagnosis of covid-19 from community acquired pneumonia. *IEEE Transactions on Med. Imaging* (2020).

32. Mei, X. *et al.* Artificial intelligence–enabled rapid diagnosis of patients with COVID-19. *Nat. Medicine* (2020).

33. Rahimzadeh, M. & Attar, A. A modified deep convolutional neural network for detecting covid-19 and pneumonia from chest x-ray images based on the concatenation of xception and resnet50v2. *Informatics Medicine Unlocked* 100360 (2020).

34. Waheed, A. *et al.* Covidgan: Data augmentation using auxiliary classifier gan for improved covid-19 detection. *IEEE Access* **8**, 91916–91923 (2020).

35. Apostolopoulos, I. D., Aznaouridis, S. I. & Tzani, M. A. Extracting possibly representative covid-19 biomarkers from x-ray images with deep learning approach and image data related to pulmonary diseases. *J. Med. Biol. Eng.* **40**, 462–469 (2020).

36. Ucar, F. & Korkmaz, D. Covidiagnosis-net: Deep bayes-squeezenet based diagnosis of the coronavirus disease 2019 (covid-19) from x-ray images. *Med. Hypotheses* **140**, 109761 (2020).

37. Ozturk, T. *et al.* Automated detection of covid-19 cases using deep neural networks with x-ray images. *Comput. Biol. Medicine* **121**, 103792 (2020).

38. Toğaçar, M., Ergen, B. & Cömert, Z. Covid-19 detection using deep learning models to exploit social mimic optimization and structured chest x-ray images using fuzzy color and stacking approaches. *Comput. Biol. Medicine* **121**, 103805 (2020).

39. Pereira, R. M., Bertolini, D., Teixeira, L. O., Silla, C. N. & Costa, Y. M. Covid-19 identification in chest x-ray images on flat and hierarchical classification scenarios. *Comput. Methods Programs Biomed.* **194**, 105532 (2020).

40. Asnaoui, K. E. & Chawki, Y. Using x-ray images and deep learning for automated detection of coronavirus disease. *J. Biomol. Struct. Dyn.* 1–12 (2020).

41. Roy, S. *et al.* Deep learning for classification and localization of covid-19 markers in point-of-care lung ultrasound. *IEEE Transactions on Med. Imaging* (2020).

42. Kelly, C. J., Karthikesalingam, A., Suleyman, M., Corrado, G. & King, D. Key challenges for delivering clinical impact with artificial intelligence. *BMC Medicine* **17**, 195 (2019).

43. Li, M. D. *et al.* Automated assessment and tracking of covid-19 pulmonary disease severity on chest radiographs using convolutional siamese neural networks. *Radiol. Artif. Intell.* **2**, e200079 (2020).

44. Koch, G., Zemel, R. & Salakhutdinov, R. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, vol. 2 (2015).

Acknowledgements

The authors want to thank the NVIDIA Clara Deploy team for their technical support and useful discussions. They also want to acknowledge the work of Mass General Brigham Enterprise Medical Imaging team, especially Ryan J Morely, Steve A. Graham, Gordon P. Garcia, Eric M. L'Italien, and Thomas E. Pryor, and the work of Software Engineering and Machine Learning Platform Team at the Center for Clinical Data Science (CCDS), particularly Artem B. Mamonov, Mark Walters, Sean W. Doyle, Kiryl Verkhovin, Anton Rodionov, Adam McCarthy, and Min Yun for their help in developing and integrating the system into the clinical settings. The CCDS is funded in part by monies and resources from Nvidia Corporation, General Electric, Nuance and Fuji.

Author contributions statement

B.H., T.S., S.D., R.N. and K.D. conceived the work. B.H. and T.S. designed the study. B.H. and A.M. implemented the study. B.H., A.M., P.F., and M.L. drafted the manuscript. B.H., A.M., P.F., M.L., S.D., and K.A. revised the manuscript. A.I., J.G., and R.H. contributed to technical support and software development. B.H., A.M., M.L., P.F., N.A., J.K. contributed to data analysis, and interpretation. All authors reviewed the manuscript.

Additional information

Competing interests

J.G., A.I., and R.H. are employees of NVIDIA. The employer had no influence on the study and on the decision to publish. The other authors declare no competing interests.

Figures

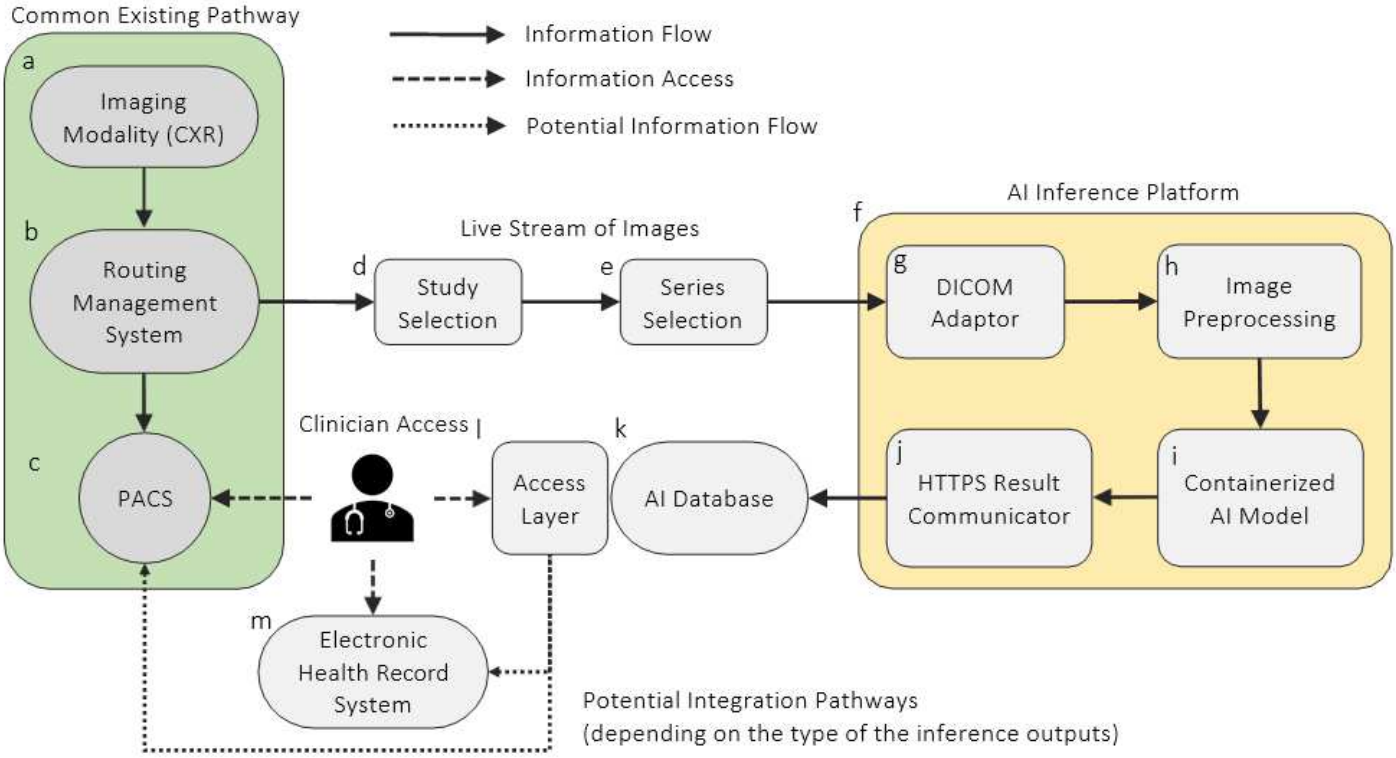

Figure 1

Workflow for clinical deployment of COVID-19 AI model

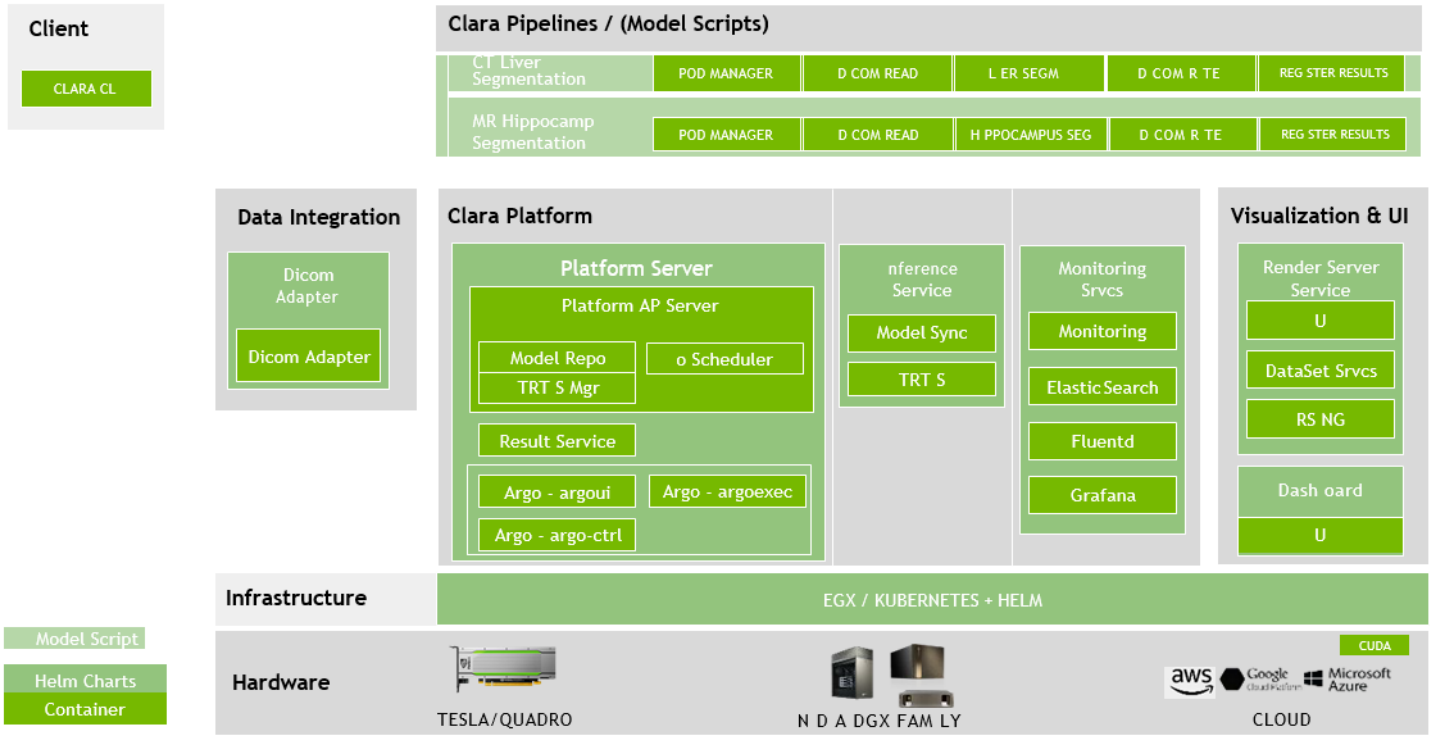

Figure 2

Clara system architecture. The Clara Deploy SDK is run via a collection of components that are deployed via model scripts, helm charts, and containers. The image is courtesy of NVIDIA.

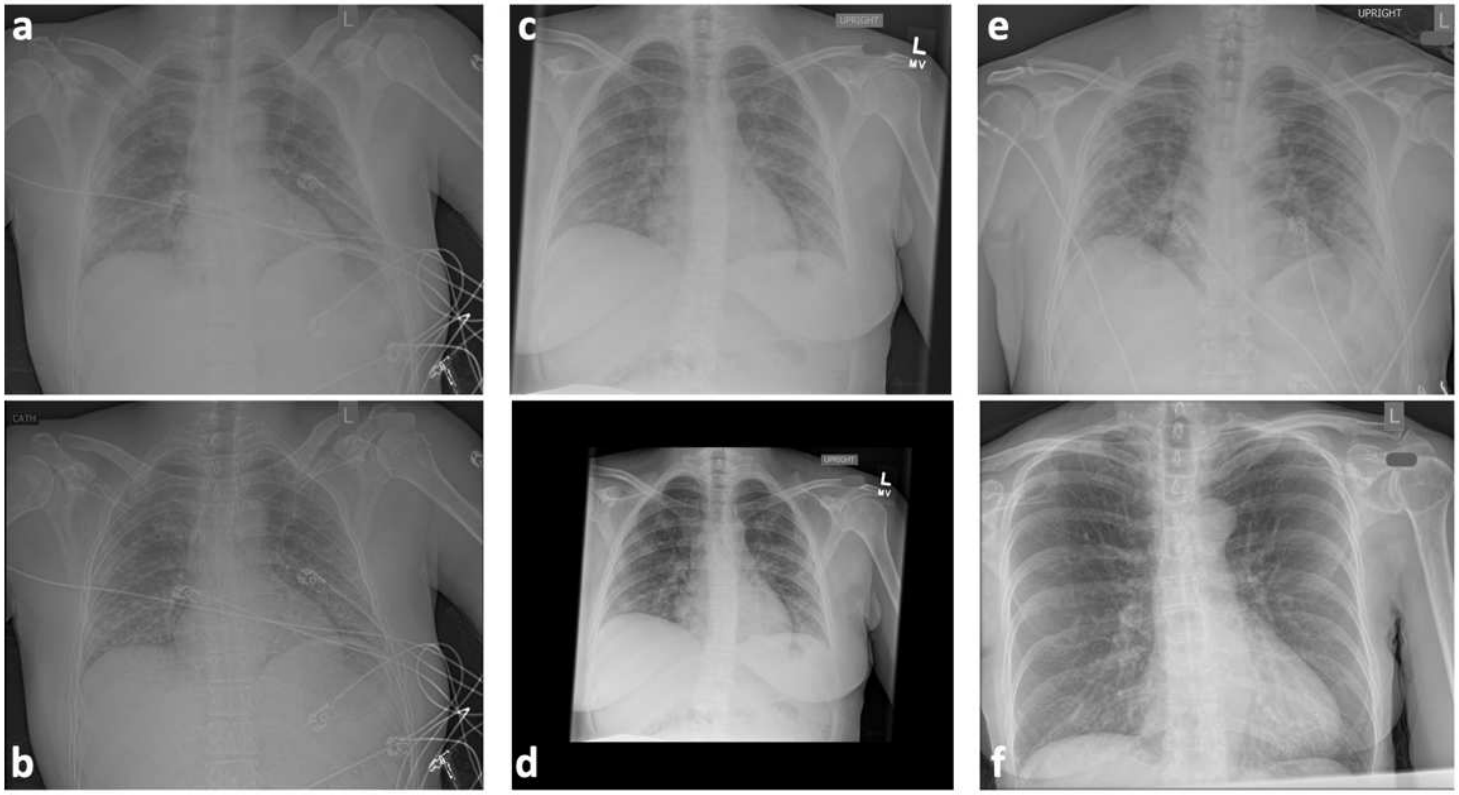

Figure 3

Examples of DICOM chest radiograph images. The top row includes acceptable model input images and on the bottom are their corresponding undesirable images from the same study. (b) is an example of CATH image (post-processed image that accentuates lines and tubes), (d) is an example of screen-shot image, and (f) is an example image from a wrong patient included with the correct image above.

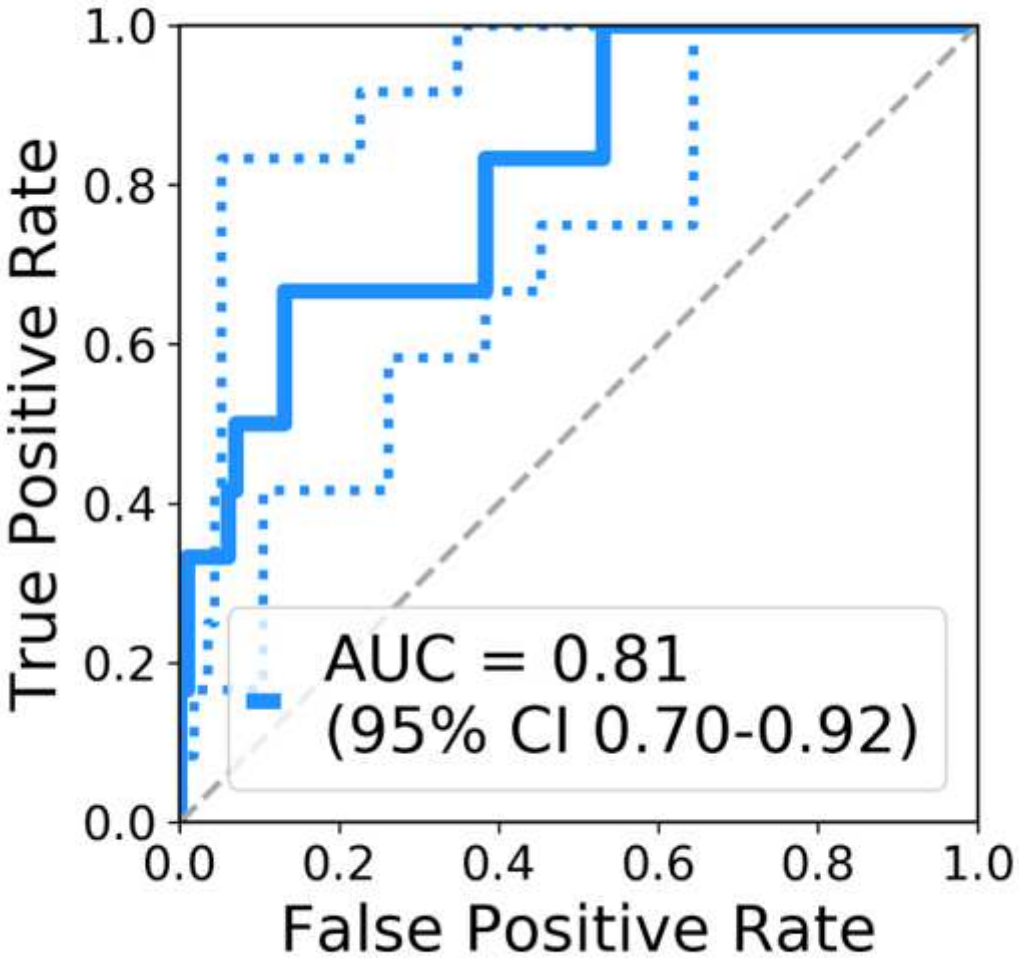

Figure 4

Receiver operating characteristic curve for prospective validation of PXS score-based (AI model) prediction of composite outcome of ICU admission/transfer, intubation, and/or death within 3 days of hospital admission for COVID-19 patients at one hospital. The PXS score (AI model output) is calculated in real-time using our AI deployment system. AUC = area under the curve; CI = confidence interval.