Developing Deep-Learning Models in the Hospital: A Case Study on the Center for Clinical Data Science

Neil A. Tenenholtz[*] **R. Winston Larson**[*] **Adam McCarthy**[*†] **Robert Keating**[‡]

Thomas Schultz[*] **Mark H. Michalski**[*]

1 Introduction

As the field of deep learning in medicine progresses from research to clinical deployment, practical considerations quickly become a primary concern for operational leadership. Hardware infrastructure, although a key enabler, presents unique challenges in the clinical arena. Required components include:

- GPU compute
- High-speed networking
- Fast storage
- Policies and procedures around usage

By stepping through the typical project workflow at the MGH & BWH Center for Clinical Data Science (CCDS), this paper explores the reasons for building such a system on-premises, the challenges that must be confronted, and a case study in how such tooling is leveraged across the project lifecycle. Our goal is to help other teams jumpstart their hardware efforts as they seek to implement deep learning in a hospital environment.

2 Background

The application of deep learning in medicine has recently garnered significant interest in the research community with a flurry of published work demonstrating the potential value of neural networks in patient diagnosis. However, there has been limited activity in implementing these algorithms in the day-to-day operations of a hospital. The MGH & BWH Center for Clinical Data Science is focused on developing deep learning models and then integrating them into clinical workflows, seeking to augment the diagnostic capabilities of clinicians. Our mission is to accelerate the application of deep learning in medicine by building products that have an impact on clinical efficiency, cost of care, and patient outcomes.

With recent advances in computer vision, our early efforts are directed at applying the latest methods to medical imaging. After this initial focus on radiology and pathology, we intend to expand into other types of clinical data, including the electronic medical record (EMR), hospital operations, and genetics. At the CCDS, our projects focus on collaborating with clinicians to build machine learning models that can have immediate impact in the clinic. Rather than replace the clinician, we seek to improve the clinician's ability to do their job efficiently and effectively – providing them with a more rich set of tools at their disposal. In radiology, this includes, but is not limited to:

[*]MGH & BWH Center for Clinical Data Science, Boston, MA USA
[†]Department of Computer Science, University of Oxford, Oxford, UK
[‡]Nvidia Corporation, Santa Clara, CA USA

- **Worklist prioritization:** Identifying critical cases to reduce time to treatment or clear beds in times of high occupancy
- **Segmenting images:** Highlighting regions of interest to improve diagnostic quality and reduce time per read
- **Calculating volumes:** Improving accuracy and efficiency in measurements
- **Suggesting report text:** Increasing efficiency by automating reporting
- **Automating mundane tasks:** Accelerating the completion of time-consuming but necessary tasks thus enabling clinicians to focus on complex and difficult cases

Our experience suggests that creating clinical impact with deep learning requires much more than a cutting-edge algorithm. Key components include:

- Clinician involvement from project onset to define high-value use cases
- Access to annotated clinical datasets
- Development of machine learning models
- Integration into the clinical workflow
- Infrastructure for model deployment
- Validation in a real-world clinical environment

Specialized hardware infrastructure is critical to success across multiple stages, including facilitating access to data, enabling the development of models, and powering the deployment and continued improvement of those models in production. Hardware infrastructure for the clinical implementation of deep learning must be able to address the compute requirements of deep learning while also satisfying the stringent standards set for production clinical systems at hospitals.

In the United States, a complex regulatory environment (e.g., HIPAA) promotes a conservative approach to managing patient data. Due to the astronomical costs of a single error, both in terms of liability and breach of patient trust, many healthcare providers are reticent to leverage third-party IT infrastructure, thus compelling them to build their own.

These challenges are exacerbated by the limited experience of most hospitals with high-performance compute infrastructure. Traditional clinical systems have limited computational demands. As a result, the high-performance GPUs, high-speed network, performant storage, and broad access patterns required for training neural networks are outside the comfort zone of hospital IT teams, which tend to specialize in high-reliability, high-uptime systems with more modest compute and data access requirements.

We have developed and continue to improve solutions to address these issues. Throughout the remainder of this paper, we will walk through a typical project lifecycle (Figure 1) highlighting the key role of hardware and lessons learned which can be applied in other clinical environments.

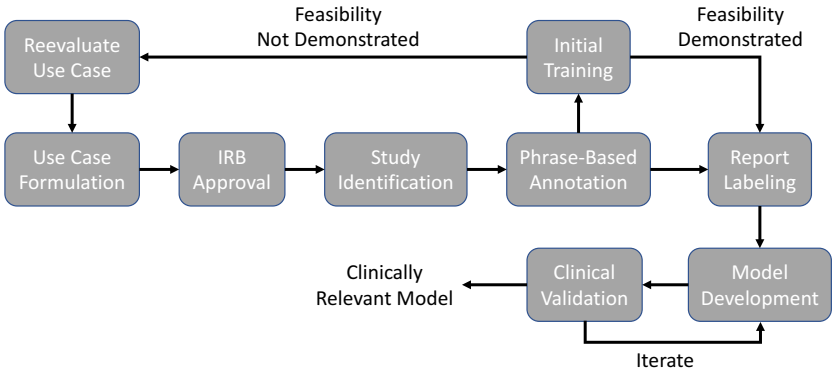

Figure 1: Discussed in detail below, the typical project lifecycle at the CCDS is predicated upon clinical feedback with continuous radiologist input and frequent evaluation on recent studies.

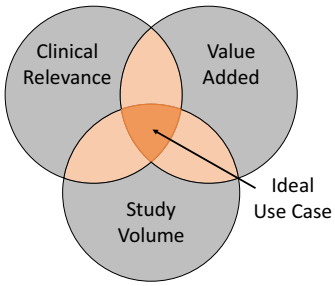

Figure 2: An ideal use case targets a task that is clinically relevant, provides value in the radiologist's workflow, and is performed with sufficient frequency to provide data for network training.

3 Model Development

3.1 Use Case Definition

Early on, we at the CCDS recognized the importance of including clinicians from project inception. The radiological workflow in particular can be unintuitive to those not well-versed in its practice – there exist a variety of important yet subtle details that can easily be missed by those who have not spent significant hours in a reading room. Therefore, to ensure the development of clinically relevant solutions, an attending physician serves as a *clinical champion* on every project, helping to define the high-level project vision, specifying product requirements, and providing radiological expertise when required.

While the technical team consisting of *machine learning scientists* and *software engineers* work directly with the clinical champion, questions often arise in day-to-day activities which do not require the specialized skill set of the clinical champion, whose time is highly valuable. To absorb some of this effort, teams also include a *clinical innovation fellow*, typically a junior physician with a quantitative background, who can provide general clinical context and serve as a product owner, providing the voice of the customer. The clinical innovation fellow also aids in project prioritization by quantifying the frequency of various radiological exams and investigating potential market impact.

Team Structure: In our experience, a team consisting of a clinical champion, a clinical innovation fellow, software engineer(s), and machine learning scientist(s) provides an efficient means of model development and deployment.

3.1.1 Protecting Patient Data

With a use-case and team established, an application is submitted to the institutional review board (IRB) to secure permission to query relevant data. Patient privacy is of the utmost concern in all CCDS projects and dictates project workflow. In addition to the complex regulatory environment surrounding the use of protected health information (PHI), at a more basic level hospitals regard themselves as the stewards of all patients' medical data and take privacy very seriously. As part of this commitment, data is not permitted to leave the institutions, including for storage or computation on cloud compute platforms. Additionally, all data is retained on servers in the data center and is not allowed to be downloaded onto local devices or workstations. Access is controlled via user permissions, which map to those listed on the approved IRB application. This compartmentalization of access ensures that an individual scientist may only view the data required to execute on his or her approved projects and also guarantees that all project team members are trained in handling patient data, a prerequisite for IRB approval.

Securing Data: The CCDS has found success in restricting data access through file-level permissions that map to individuals listed on an approved IRB application.

3.2 Study Identification and Labeling

Prior to training, a cohort must first be identified. While the use case may define a population of interest (e.g., patients who received MR imaging to quantify core size after being diagnosed with an ischemic stroke), oftentimes there is no straightforward manner to directly isolate the desired group of patients. Data may reside in closed, difficult-to-access clinical systems; may be reported in an indirect and/or unstructured manner; or may not even be permanently recorded.

After experimenting with a few different approaches, the CCDS has found success in mining radiological reports for keywords and phrases to convert the semi-structured data of the reports to labels that can be used for training. To facilitate this process, a report annotation tool was developed, which has since become the de-facto standard within our organization for the labeling of data.

A broad initial cohort is first identified by study date, to exclude the oldest acquisitions, and exam code, an identifier specifying the imaging modality, body part(s) imaged, and other acquisition parameters. The corresponding reports are then loaded into the aforementioned annotation tool for analysis. For classification tasks, the clinical innovation fellows view the reports, highlight phrases of interest, and associate them with a given label (Figure 3 - top)[1]. Any report containing this phrase, and not previously matched to another phrase, is marked with a "soft label." This process, internally referred to as annotation, allows the innovation fellow to rapidly cull vast numbers of reports and group them together, leveraging the semi-structured templates often employed by radiologists.

Because certain phrases are deemed higher priority than others, a second screen is offered where the innovation fellow can optionally re-prioritize phrases by dragging rows in the table (Figure 3 - middle). Phrases listed above others are applied first in the soft-labeling process. This screen also allows for the editing/removal of phrases (error correction) and a fine-grained option to view all reports matching a given phrase to confirm the phrase is being applied as expected.

These annotations are understood to introduce a degree of error, which can vary from project to project based on the level of structure found in the report for a given exam type. However, the speed in which they can be produced allows for the low-cost de-risking of projects. An initial proof-of-concept model successfully trained on these noisy labels provides an early indicator of project feasibility. It also enables the parallelization of labor – while training is ongoing, the more resource-intensive verification of annotations (described next) can be underway.

Annotations are then manually confirmed or corrected on a study-by-study basis (Figure 3 - bottom). By previously grouping the reports, they can now be rapidly processed with minimal mental burden, increasing efficiency. This time-intensive task is largely performed using a low-cost annotation pipeline consisting of approved hospital affiliates under the supervision of innovation fellows. Affiliates are assessed both via inter-rater agreement and against a small set curated by the innovation fellow. Because of the imbalanced nature of some datasets, it is beneficial to allow the labelers to preferentially select the likely study outcome; this is accomplished using the label associated with the previously established phrase.

Labeling Data: By first applying phrase-based annotations, an initial set of low-cost training labels can be provided to the machine learning scientists enabling project exploration. Due to the level of error present in phrase-based annotations, subsequent label confirmation and/or correction should be performed prior to the completion of model development.

3.3 Image Preprocessing and Initial Model Development

Once the studies have been annotated, early model development may commence. The first step in this process is the conversion of studies to an easy-to-use file format. Images are copied from the clinical PACS via a research vendor neutral archive to minimize the risk to any clinical systems and are placed in a directory on a network storage solution with permission restricted to those listed on the approved IRB application. Volumetric data (e.g., MR, CT, etc.) is often converted from DICOM, the standard medical imaging format used by PACS in which each slice is represented by a separate file, to NIFTI, a file format in which the entirety of a series is represented within a single addressable volume.

[1] Analogous processes are followed for other types of problems.

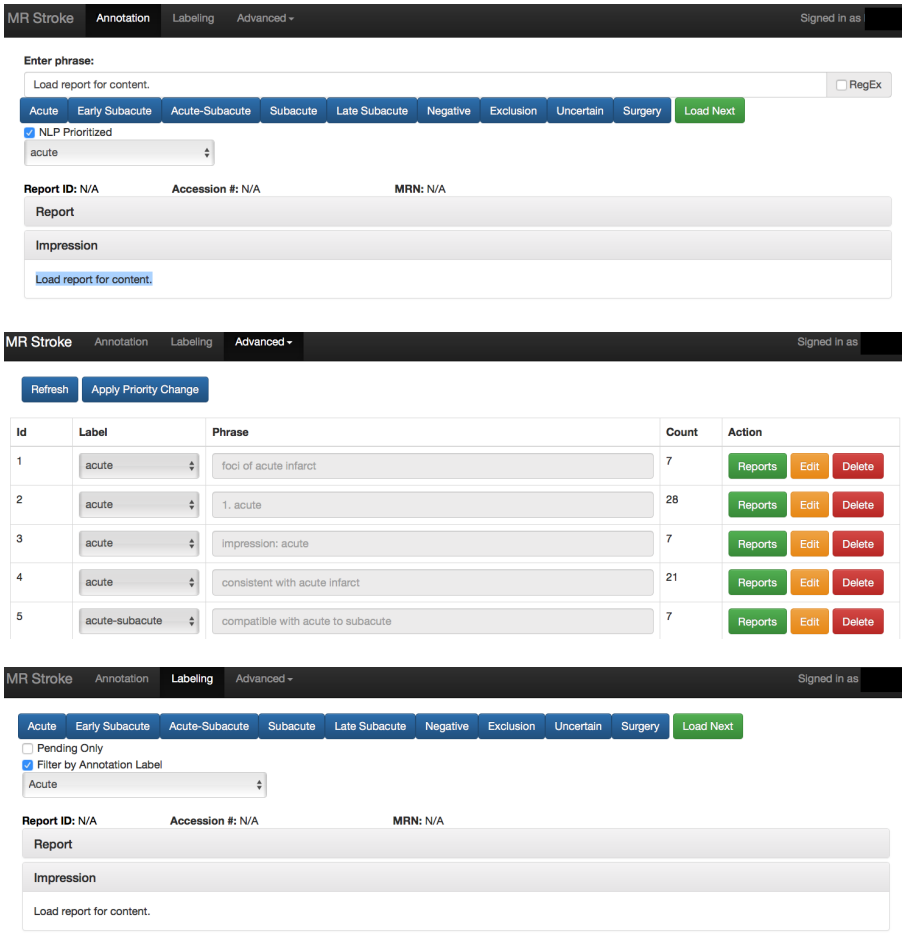

Figure 3: The labeling of studies from radiology reports was performed via a custom-developed web application. Soft labels were first assigned via phrase-based matching (top). Phrases were then reprioritized and assessed for quality (middle). Finally, manual confirmation of soft labels was performed on a study-by-study basis (bottom).

With the imaging data now available, the studies must be appropriately normalized. While the pixel/voxel intensity of some modalities is a physical property of material being imaged (e.g., the Hounsfield Unit in CT), others such as MR are machine and acquisition setting dependent. Additionally, image resolution and slice thickness can vary dramatically and must be accounted for. While the precise steps in this process can vary from project to project, the development of a normalization pipeline is almost always highly interactive, thus necessitating the need for an interactive session. Once complete, normalization of all studies is performed in a distributed fashion using a workflow engine (e.g., Apache Airflow[2]). Development of operators is a task shared by the machine learning scientists and software engineers, many of whom have extensive medical imaging experience and offer expert-level knowledge in various modalities and the complex DICOM format.

The initial stages of model development also follow a largely interactive workflow. In these interactive sessions, models are developed and trained for a few epochs to ensure functional correctness. Because of PHI concerns, these interactive sessions must be launched remotely in the Partners' data center to ensure that no PHI is stored locally on easily removable hardware. This environment is designed to serve as a scaled-down version of the CCDS compute cluster (described later). Therefore, each machine learning scientist is allocated two to four high-performance GPUs (Nvidia `Tesla P100` or `Tesla V100`), which support `GPUDirect P2P` for efficient intra-node GPU communication and `GPUDirect RDMA` for inter-node communication. These features are highly advantageous

[2] `https://airflow.apache.org/`

when training models on volumetric data, which have proven to be both highly compute- and memory-intensive. The 16GB of high-speed HBM2 memory, support for half-precision floating-point operations, and `TensorCore` mixed-precision matrix multiply/add (Tesla V100 only) greatly reduce the hardware required relative to consumer GPUs. These benefits are reflected across the CCDS's infrastructure. While high performance is not necessarily required in this early model development stage, these features are heavily leveraged during cluster jobs and therefore must be available in the development environment to ensure model correctness.

Two approaches are currently being explored to support this workflow:

1. **Static hardware allocation:** Each machine learning scientist is provided with a dedicated machine, either physical or virtual, on which all exploration of image normalization techniques and initial model development can be performed.
2. **Dynamic hardware allocation:** Nodes are allocated from a high-priority queue via the cluster's scheduler. Requests by an individual for a second node are highly deprioritized relative to the first request.

The tradeoff between the two is largely one of resource utilization and friction. With static hardware allocation, the scientist's environment is trivially maintained with no additional effort required to ensure portability between sessions. However, during times when the system is not in use, the GPUs remain idle and could potentially be allocated to other scientists. Conversely, with dynamic hardware allocation, although the issue of resource utilization is largely eliminated by the cluster's scheduler, care must be taken in terms of environment management to ensure environment changes persist across sessions. While Docker[3] solves much of this problem, certain frictions are largely unavoidable.

Hardware for Model Development: Volumetric data, common in medical imaging, can prove to be taxing on consumer-level hardware. The high memory capacity, accelerated inter-GPU communication, and reduced precision floating-point operations of the Tesla P100 and V100 are able to mitigate some of these challenges. The CCDS has had a positive experience initially developing models on a small number of these GPUs and then later scaling to larger quantities.

3.4 Model Training at Scale

Once a set of candidate architectures are identified, they are trained at scale using the CCDS's compute cluster. While leveraging the same hardware, these operations are largely performed in two steps:

1. **Hyperparameter Search:** The candidate architectures are tested with a broad array of hyperparameter configurations to identify the optimal model configuration. Depending on the scientist's preference, this may either be dictated by a random search or Bayesian Optimization. By leveraging the excess capacity of the cluster, large numbers of configurations can be tested in parallel, converting the formerly serial task of testing a variety of architectures and configurations into a parallel one. This has enabled our organization to quickly iterate and optimize.
2. **Large-Scale Training:** Once a limited set of model architectures and hyperparameter configurations have been identified, each model is trained to convergence, seeking to identify the best model in the cohort. Successful large-scale training relies on parallelizing the model across many GPUs with efficient inter-node communication.

We have designed our cluster to accommodate the needs of this workflow. Compute nodes reside behind IBM's LSF scheduler, which delegates submitted jobs to available resources and ensures fair distribution of nodes. Jobs are submitted via Docker containers to manage development environments and ensure consistency, simplifying the management of the cluster and the number of packages installed on each node. `nvidia-docker` allows for the seamless integration of GPUs into the container with the recent 2.0 release further reducing frictions. Additional benefits realized by the CCDS include easy selection of a Tensorflow release, which often requires a particular version of

[3] `https://www.docker.com/`

Figure 4: The CCDS team recently received the world's first Volta-based DGX-1 system.

Nvidia's highly optimized `cuDNN` library; flexibility in the selection of base containers, including non-Nvidia containers if desired; and a simple means of GPU isolation.

With an easy-to-use containerized environment, the CCDS is readily able to parallelize jobs across multiple nodes and GPUs leveraging TensorFlow's transparent synchronization operations and custom in-house libraries. We also heavily rely on Nvidia's `NCCL` library, which is integrated into the framework, for efficient multi-GPU operations. This tooling allows us to reduce training time and shorten our model development cycles.

As briefly touched upon earlier, it is during large-scale training where the benefits of the investment in cluster infrastructure are realized. In these distributed jobs and especially when working with volumetric datasets, the choice of GPU and optimized inter-node communication become critical. Volumetric data from CT and MR scans, the latter of which will also often include multiple acquisition sequences and thus multiple channels in the input volume, require large amounts of GPU memory. While a greater number of GPUs with a smaller amount of memory could be utilized, this would place an additional reliance on the slower inter-node network, increasing training times. Although the 200 Gbps offered by Infiniband EDR provides fast, low-latency inter-node connectivity, it is significantly less performant than the intra-GPU HBM2 memory (732 GBps) and the intra-node, inter-GPU `NVLink` connection (160 GBps) offered by the `DGX-1` Pascal-based systems installed at the CCDS. The Volta-based DGX-1, also installed at the CCDS (Figure 4), demonstrates even greater performance with memory and NVLink speeds of 900 GBps and 300 GBps respectively. When combined with the software optimizations the DGX-1 offers, our scientists are able to successfully train models with shorter turn-around times.

Cluster Infrastructure: The high-performance DGX-1 provides a powerful compute platform. When connected together with high-speed Infiniband, one can efficiently train large models with reasonable batch sizes on volumetric medical data.

4 Clinical Validation

Clinical validation of models and tools is a crucial step in our development process. In an academic setting, a model is considered successful if it is able to outperform three to four radiologists on a test set. Because we are focused on building tools that will be used by clinicians to diagnose patients, we are creating a rigorous validation process for ensuring our models will be clinically viable.

4.1 Pre-Deployment Validation

Validation of our models begins during model development. We build our cohorts and training sets in collaboration with clinicians. Together, we collect a robust training set that includes more than just the clean, ideal images that are positive or negative for a given condition. We also ensure that we account for lower quality studies (e.g., movement on the scanner or image artifacts) as well as studies

that are considered more "difficult" to read (e.g., mimics, atypical anatomy, and post-surgical follow up). Through this process, we train our models to account for data that will be encountered in the reading room on an everyday basis.

To further stress test our models, we often assess them on consecutive studies newly acquired from the hospitals' scanners. Because large numbers of images are being acquired every day, we are able to continually test our model throughout the development lifecycle. This highlights any instances where our validation and test sets may not be representative of the studies acquired in the present-day imaging suites and ensures all models perform well on the hospitals' current scanners. It also evaluates model performance in the highly unbalanced datasets that are commonly found in a clinical setting. This pre-deployment validation de-risks the integration process and provides a strong signal to both the CCDS and the clinical champion when the model is ready for deployment.

De-risking a Model: A large medical center will provide a continuous stream of images acquired under both ideal and suboptimal conditions. It is important to evaluate a model in these real-world settings to identify any potential issues prior to clinical deployment.

4.2 Clinical Integration

Once model performance has been validated and the clinical champion has indicated support for reading room integration, the process of clinical integration begins. This largely occurs in two phases: software integration and release to production.

The process is initiated by the machine learning scientist handing off a dockerized model, which encapsulates all dependencies, to the software team. The software team and UI/UX designer then work to execute the vision of the clinical champion and integrate the model into the clinical workflow. Such efforts are often challenging due to the black-box nature of most clinical solutions. Not designed to interact with other processes, these products do not offer a straightforward path to inject content. Therefore, the developers must be creative, oftentimes leveraging low-level Windows APIs to ensure synchronization of content and mimic user interaction. Given the complexities of such a design and the associated maintenance challenges, the team will often face tradeoffs and must negotiate a balance between achieving the originally envisioned product and speed of development. These difficulties are navigated by directly observing the clinical workflow, discussing proposed alternatives with the clinical champion, and settling upon a resolution. Once finalized and tested locally, the software is evaluated by the clinical champion on a test platform, which mirrors the production infrastructure.

Tested and bundled for deployment, the solution is then pushed to a canary server and used by the clinical champion in a production environment. Upon clinician approval, the solution is next rolled out to a set of radiologists who have elected to use the newest tooling. Finally, after validating performance in the canary environment, the software is rolled out to all radiologists in the reading room.

To support this software, models run within virtual machines, the preferred means of deployment for the hospital's IT team. This allows for a smaller hardware footprint and simple rollover to other servers in the event of a hardware failure. While this has traditionally been the modus operandi for software deployments within the hospital, the need for a GPU at inference introduces additional complexities. To solve this challenge, the team has elected to leverage the Nvidia `Tesla P40`, which possesses 24GB of RAM. When coupled the Nvidia `Virtual GPU` software, each GPU can be split into a number of virtual GPUs due to the lower memory requirements at inference. This allows each VM to leverage a partial GPU, thus minimizing the hardware footprint.

Conforming to Existing Workflows: To achieve buy-in from all stakeholders, it is important to insert oneself into existing clinical and IT workflows. Given the black-box nature of most clinical systems, software integration presents a unique set of challenges that must be solved creatively. Virtualization is a common solution within hospital IT to ensure security and high availability, and therefore the Nvidia Virtual GPU software implemented on a minimal hardware footprint (e.g., an array of Tesla P40s) provides a solution that seamlessly fits in existing IT workflows.

4.3 Post-Deployment Validation

After integration with the hospital's clinical systems is complete, the model is evaluated in the clinicians' day-to-day operations. This process helps us to evaluate:

1. **Model performance:** Does the model perform well in the reading room, meeting the expectations of radiologists?
2. **Usability:** Does the model and its user interface increase the effectiveness and efficiency of the clinical workflow?

We test both the performance of our models and the usability of our tools in a highly collaborative and iterative process with our clinical partners. Our software and user-interface developers continue to observe clinicians in order to understand how the tool has been adopted across the reading room. With subtle variations in workflow from clinician to clinician, changes are made to optimize usability for the department rather than a particular radiologist. This increases the likelihood of the model improving rather than inhibiting clinician performance and also helps drive adoption. With more radiologists using the tool, more feedback is acquired, both in terms of model performance and user workflow, enabling the team to further improve the model, creating a virtuous cycle.

Achieving buy-in is crucial as model validation remains a moving target. Scanners, their sequences, their image resolution, and their reconstruction algorithms are constantly changing, and the team is not always alerted to these software and/or hardware upgrades. Therefore, continuous monitoring is required to ensure there is no degradation in model performance. While manual feedback loops could be employed, such procedures are error-prone and add additional responsibilities to the radiologist's workload. To eliminate this dependency and minimize clinician burden, we have automated this process; all model output is logged alongside the radiologist's final report. Analytics can then be run to assess model performance over time and flag notable changes. Such challenges will remain even after commercialization and thus have attracted the attention of the American College of Radiology, which intends to help support this process.

Continuous Validation: Given the dynamic imaging landscape, model output must be monitored to ensure continued performance. An automated monitoring platform provides a robust solution that can alert the algorithm developers to any change in real-world performance.

5 Conclusion

Given the potentially significant impact of downtime on patient care, hospitals tend to be conservative in the adoption of new technologies. It is therefore crucial that any new solution be thoroughly validated prior to integration and highly beneficial for it to conform to existing workflows. While the rise of deep learning medicine has brought to the forefront a host of new challenges, we have found that with an appropriate combination of creativity, vigilance, and careful selection of vendor solutions, these difficulties can be overcome and the full weight of deep learning can be brought to bear in a clinical environment.