International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

Cloud-native data warehousing for large-scale medical imaging analysis using deep learning in US hospital

Amit Nandal

Independent Researcher, (MBA, Master's Computer Information Science, ITIL) PA, US

Email: nandalamit2@gmail.com

Abstract

New possibilities for automated evaluations and information extraction from massive collections of photographs have never existed before, thanks to the fast development of AI in imaging for medical purposes. It may be challenging to meet the computing needs of such cutting-edge AI technologies with on-premises capacities. Cloud computing promises very cheap access and flexibility. Using the cloud for healthcare image processing is an area where there is a dearth of research on the cost-benefit analysis. We explore the possibility of augmenting the National Lung Screening Trial (NLST) Computed Tomography (CT) images accessible via the NCI Imaging Data Commons (IDC) with artificial intelligence (AI) through the utilisation of computational resources provided by the cloud. To automate segmenting of images using TotalSegmentator and pyradiomics feature extraction for a massive cohort including over 126,000 CT volumes from over 26,000 consumers, we assessed the NCI Cancer Research Data Commons (CRDC) Cloud Resources - Terra (FireCloud) and Seven Bridges-Cancer Genomics Cloud (SB-CGC) platforms.

In contrast to the predicted 522 days required on a single workstation, we were able to finish analysis in under 9 hours by using over 21,000 virtual machines (VMs) during the calculation. Using the cloud for this study came to a total of $1,011.05.

Keywords: Cloud Computing,Medical Imaging, AI Segmentation, Data Curation,Scalability

1.Introduction

Researchers in the field of medical imaging are being faced with datasets that are becoming larger by the day. Clinical trials and other specialised data gathering and sharing initiatives are potential sources for such datasets. Annotating the areas of interest and post-processing to obtain quantitative information1 are common steps in gaining insights from the photos.

Quantitative measures have several potential uses; for example, they may be used to assess potential imaging biomarkers, integrate imaging data with other disease-related sources for multi-omics analysis, and perform population studies2. When dealing with massive datasets, manually annotating the pictures' areas of interest is often not feasible. Due to variations in reader training, standards defining the segmented area, and inter-reader variability, even professional annotations might have shortcomings. Recent developments in automated annotation technologies have raised the possibility that AI-based curation might be a useful tool for annotating large-scale images. The effective application of these automated approaches to large datasets,

however, is still an important obstacle. This research delves at the utilisation of computing resources in the cloud and how they may open up access to large-scale annotation powered by artificial intelligence. Among the most extensive publicly accessible cancer screening imaging datasets is the National Lung Screening Trial (NLST)3,4. Over 26,000 patients' CT scans, totalling more than 10 gigabytes, are stored there. More than 30 cancer centres used scanners from four different manufacturers to compile the screening CT pictures that make up the NLST collection, which is rather diverse. NLST pictures come with extensive clinical information, which includes features that describe imaging results (such as the CT slice with the largest abnormality width). This metadata allows for the examination of several secondary hypotheses, as shown, for example, by Zeleznik et al.5. Volumetric categorisations of anomalies or anatomical structures are missing from the original NLST data, which restricts their use for subsequent analysis. As mentioned earlier6, volumetric categorisations of the relevant areas (organs and cancers alike) greatly enhance the database's utility. This is because these division are a typical preliminary processing process in the more involved evaluation pipelines that look into the imaging biomarkers' utility in different contexts, and they can be utilised to extract a variety of pictures-based features. With segmentations at our fingertips, we can test hypotheses linking clinical results to picture content, enhance data searchability, and construct cohorts according to the presence of certain architecture in images. Applying an algorithm for segmentation to a big dataset also allows one to test its resilience and generalisability. The National Cancer Institute's Imaging Data Commons (IDC)7 just made NLST public. Using the elastic cloud resources, IDC hopes to simplify content analysis by storing all of its data in the cloud.

The major contributions are:

1) an in-depth investigation of the various trade-offs involved in optimising cloud infrastructure for large-scale image evaluation;

2) CloudSegmentator, a reusable and extensible open-source framework to implement the established workflows;

3) practical suggestions for taking advantage of the cloud for medical image computing tasks on a large scale. Additionally, we provide the analysis's outcomes, which include 9,565,554 division of the underlying anatomical framework and their corresponding radiomics characteristics in IDC as of version 18.

A new Deep Learning (DL) model called TotalSegmentator8 can separate up to 104 different anatomical structures (excluding tumours) from CT images. We show in this research how to apply Total Segmentator on the NLST collection using cloud computing resources. It would take more than a year to sequentially apply the established process to NLST on a typical workstation, considering that the NLST collection contains more than 200,000 CT images. Many laboratories provide GPU clusters and other parallel computing technologies that associated researchers may use for free or at a modest fee. However, there is a great deal of variation

in the availability, performance, and system configurations of institutional computing resources. They may be expensive to maintain, need regular hardware updates to stay current, and may cause friction amongst lab groups when it comes to sharing computing resources. Data centres and support staff are being reevaluated by several institutions that have run their own HPC installations in the past or at present. It is highly motivated to have a deeper grasp of the possibilities for scalable and reliably accessible cloud resources in order to speed up the study due to these concerns. In a cloud environment, anyone with enough money may run the analysis, regardless of whether they have the necessary authorisation from the owners of any institutional resources or whether their installation settings and hardware are consistent with TotalSegmentator's specifications. In addition to improving cancer imaging data with anatomical structure annotations, the advanced AI application TotalSegmentator may be used as a stand-in for evaluating the efficiency and cost-effectiveness of cloud computing. While we anticipate that our study's per-segment radiomic feature quantitative data, in conjunction with NLST clinical data, might be used to investigate cancer research ideas, that particular sort of analysis is not addressed in this work.

2.Methodology

2.1 The processing pipeline for TotalSegmentator

With the newly released TotalSegmentator8 version 1, built on the nnU-Net framework 10, 104 anatomic structures (27 organs, 59 bones, 10 muscles, and 8 arteries) may be segmented from CT scans. The datasets used for instruction and assessment the simulation were obtained from the Medical Centre of the University Basel. The set used for training contained 1024 CT volumes, while the final assessment information set contained 4004 volumes. The information sets included scans from people suffering from various diseases, and the majority of the images were gathered utilising the same the company's machinery. Initial testing of the TotalSegmentator model showed that it achieved good accuracy when segmenting on the training dataset. Researchers looked for age-related relationships between CT attenuation and segmentation structure volume in the study population. The framework is accessible on GitHub and includes a command-line utility for applying it to DICOM or NIfTI images.

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

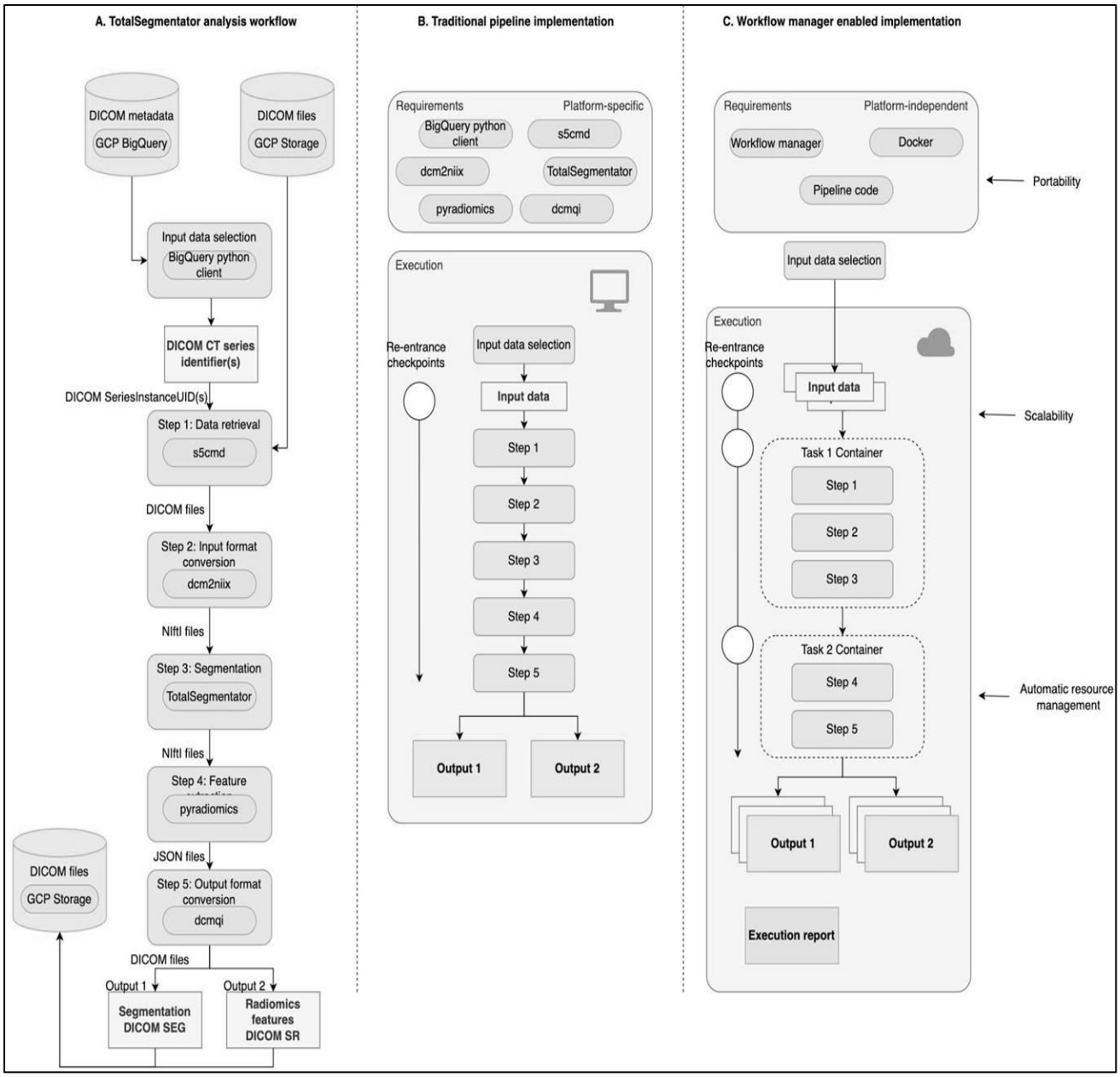

Figure 1:An overview of the deployment of the Total Segmentator analytical process

Figure 1 also shows the following procedures for applying TotalSegmentator on the IDC NLST gathering:

The proposed methodology for cloud-based large-scale curation of medical imaging data by AI segmentation is a multi-stage pipeline. Medical imaging datasets (CT, MRI, X-ray, etc.) are uploaded to the cloud with secure storage and compute capabilities. Following this, segmentation models driven by AI—typically deep learning architectures (U-Net or Transformer based)—are applied to the image to automatically identify and label anatomical structures or pathological regions. The models will be deployed in container-based environments for reproducibility and parallel processing (Docker-Kubernetes). The results are

validated either manually by expert radiologists or through semi-automated quality-check algorithms. The curated and annotated dataset will be stored in structured formats such as DICOM or NIfTI and indexed with metadata for easy retrieval, future analysis, and model training. Concern for data privacy, data security, and regulations related to medical data (such as HIPAA or GDPR) will be protected all throughout the pipeline.

1.Uploading Secured Data to the Cloud

The medical imaging datasets (CTs, MRIs, X-rays) are uploaded securely to a cloud where it can be scaled up for storage and computations. During this setup, it is ensured that the large quantity of data is satisfied in an efficient and secure way.

2.AI Segmentation

Several advanced artificial intelligence models using deep learning architectures ranging from U-Net to Transformer-based networks are being deployed on imaging data. These models identify and label anatomical structures or pathological regions to allow fine segmentation of the images.

3.Containerized Deployment

Segmentation model serves in a containerized environment, using tools like Docker and Kubernetes. In doing so, this provides reproducibility and parallel processing on many virtual machines, thus increasing scalability.

4.Validation and Quality Control

The segmented output is rigorously validated by expert radiologist review or semi-automated quality control algorithms. This step is crucial for ensuring the accuracy and reliability of the segmentation results.

5.Curation and Storage

Curated datasets are stored in structured formats such as DICOM or NIfTI, and are indexed with complete metadata. Such organization allows for easy retrieval for future analysis, model training, or even further research. Data Privacy and Regulatory Compliance

Strict measures are applied at all times through the entire pipeline to maintain data privacy and security, as well as compliance with regulations such as HIPAA or GDPR. This ensures that sensitive medical information is protected at every stage of the processing.

1.Choosing the data to enter Several acquisitions are common in CT scans; they might involve pictures that are not appropriate for volumetric evaluation because to projecting or localiser. Additionally, it may be impossible to rebuild a few CT scans into coherent 3D representations owing to factors such as insufficient or damaged data. With TotalSegmentator, you can only expect a subset of CT series to work.

2. Data Recovery Obtaining the DICOM data from the IDC cloud buckets that correspond to the CT series under analysis should be done effectively. Converting the input source The original DICOM format of the NLST dataset was used for sharing. Despite TotalSegmentator's capacity to accept properly structured CT volumes straight from DICOM, we opted to use dcm2niix12, a more reliable and fault-tolerant conversion tool, and feed TotalSegmentatorNIfTI files in a uniform

format to accommodate the dataset's inherent variability.

3: Separation To run TotalSegmentator on an adequate volumetric CT series, you'll need a powerful enough graphics processing unit (GPU) or central processing unit (CPU) and enough RAM.

4: Extracting Radiomics Features To help find failed instances and outliers, fundamental radiomics features (such sphericity, mean diminution, or volume) may be used to summarise the subdivisions' basic properties. Radiomics characteristics are also useful for secondary NLST collection analysis, like the one done by Wasserthal et al.13, which may look at acquisition-, age-, or sex-related trends.

5: Converting Output Formats In order to ensure that the results can be easily shared and visualised with IDC tools, we transform them to DICOM Segmentation layout. This allows us to achieve FAIR visualisation of the information for archival objectives, as well as extracting metadata that is needed for exploring and finding the results of the analysis.Considering the processing stages mentioned earlier and the massive size of the NLST gathering, there is a compelling need to parallelise analysis in order to decrease processing time. Using assets retrieved from the cloud, we tackle this difficulty in our research.

2.2 Statements on ethics

Computerised tomographic scans of human participants from the National Lung Screening Trial database (NLST)3,4,14 were analysed in this work. Because the study used de-identified publically accessible information it is eligible for NIH Exempt Human Subject Research under Exemption 4.

2.2.1Terra and SB-CGC: Cloud Infrastructure for Scalable Bioinformatics Workflows

Terra is a managed bioinformatics service born out of a collaborative effort between the Broad Institute, Microsoft, and Verily. Terra was created for the express purpose of facilitating large-scale analysis of biomedical data and has gained traction through partnerships with a series of NIH organizations. These partnerships led to branded versions special to specific institutes such as FireCloud (NCI), AnVIL (NHGRI), BioData Catalyst (NHLBI), and eLwazi (NIBIB). Terra workflows are written in Workflow Description Language (WDL), a domain-specific language for defining computational pipelines. A WDL script contains input declarations, virtual machine (VM) resource specifications, Docker containers to encapsulate task environments, executable commands, and expected outputs. The multiple tasks within a workflow are, therefore, capable of executing independently and in parallel across separate Dockerized VMs, maximizing compute efficiency for Terra. Terra allows parameterized execution with the aid of data tables that set the input references and are later populated post-execution with output file links. Cromwell acts as the execution engine for WDL scripts within Terra; it manages the execution of the tasks over the Google Cloud Platform (GCP) using the Google Cloud Life Sciences API. Although GCP stands as the primary cloud backend of

Terra, Microsoft Azure is now made available for public preview, thereby indicating that the WDL standard is thereby cloud-agnostic. Workflows can be customized by users according to CPUs, Memory, and GPU desired configuration. The specification of WDL is being cared for by the open-source community of OpenWDL, assuring the portability across cloud environments and local clusters. WDL syntax is reasonably to-the-point and devoid of unnecessary complexities but does illuminate the expressivity in the context of bioinformatics workflows.

2.2.2 SB-CGC with the Ecosystem of Seven Bridges

The SB-CGC is a cloud-based research platform that was built by Velsera on an NCI contract. It runs bioinformatics pipelines defined in CWL format. The Common Workflow Language, CWL, describes how to define and run workflows with data analysis. Such as WDL, CWL aims for portability, reproducibility, and containerization.

A CWL workflow consists of:

- Inputs and outputs
- Execution through a Docker task
- Ian runtime description memories such as memory and cpu

Cwl has support for both linear and dispersed workflows so tool definition. Terra is otherwise for customization of vm, N1-not available on the SB-CGC platform-the resource offered only utilizes preconfigured aws virtual machine types - specifically from the N family-and they do not support gpu-enabled instances. Cloud or locationally installed, SB-CGC workflows were designed to allow users' flexibility depending on their infrastructure. In addition to its NCI role, SB also partners with other initiatives:

- CAVATICA with Children's Hospital of Philadelphia
- BioData Catalyst at NHLBI

These partnerships demonstrate SB's commitment to establish and contribute interoperable, cloud-based platforms for the large-scale biomedical research within the NIH ecosystem of data.

2.2.3 Implementation and enhancement of the analytical process on CRDC Cloud Infrastructure

After the workflow's functioning prototype was finalised in a Jupyter notebook, we released it to the Terra and SB-CGC platforms for testing purposes find all of elements mentioned in this section within the matching CloudSegmentator software library.We started by creating Docker images that included all of the tools that would be required for the process. We created many picture versions, each including a unique subset of the process phases. The purpose of this was to determine how much it would cost to schedule certain process activities using different GPU and CPU virtual machine configurations. To use the GPU, a job must have TotalSegmentator inference. However, more data transfers and scheduling overheads would be incurred if individual activities of the same workflow instance are scheduled across various virtual machines. Virtual machines (VMs) that may be halted by the hosting service at any moment depending on use demand are called preemptible or spot VMs, and they are offered by most cloud providers at a

much lower price than conventional VMs. If you try to run everything on a single preemptible virtual machine (VM), you'll end up with a higher total uninterrupted time requirement and a higher interruption probability. In order to experimentally evaluate these alternatives, we created several process setups. Docker images and Python notebooks were created for each of the following workflow setups in CloudSegmentator: cloud segmentation, total segmentation, workflows, and notes.

Figure 1B shows oneVM, an approach similar to a conventional pipeline in which all process phases are run on a single virtual machine.

2. Two-VM (as shown in Figure 1C): the processes involved in the workflow are divided into two parts: 2) radiomics feature extraction and conversion of the inference findings (steps 4-5 of the pipeline), and 3) downloading, converting, and executing TotalSegmentator inference on the DICOM images (steps 1-3 of the pipeline). Two virtual machines (VMs) may run tasks simultaneously.

3. three-VM: the process is divided into three steps: 1) format conversion and input download, 2) inference, and 3) extraction of radiomicsfeatures.After that, we defined the analysis workflows using WDL for the Terra platform and CWL for the SB-CGC platform Workflow definitions in WDL and CWL are conceptually similar in that they both consist of at least one job. When working with multi-task workflows, Terra/SB-CGC may intelligently pass on the results of one action to the next. The following elements are included in the description of every job:

1.Runtime: the virtual machine (VM) environment (CPU, GPU, RAM, memory, storage, etc.) needed to perform the job, as well as the Docker image to be used for creating the container to execute the task. Preemptible virtual machines (VMs) may be prescribed to run in either SB-CGC or Terra, and they can be terminated (preempted) if needed. If a task is preempted, SB-CGC will always resume it with a conventional, non-preemptibleVMs, but Terra may restart it with as many preemptible VMs as the configuration option allows.

2.Second, a command is a set of instructions that specify how to carry out a certain task's processing. In our scenario, a Python notebook defines each job, and Papermill is used to run and parameterise it.

3. The input and output variables are used to provide parameters for a task's execution and to capture its outcomes.

The workflow specification allows for control over the choice of CPU or GPU platform for task execution, making it easier to examine trade-offs linked to GPU or preemptible VM utilisation.

2.2.4 Things to think about while executing workflows on the cloud

The complexity and adaptability of process execution in the cloud are enhanced by its extensive configuration options. Predicting, limiting, and minimising the costs of executing workflows in the cloud is a vital capability. We go over some of the main configuration settings that affect cloud charges in this section. Although most of these insights may be applicable to other cloud providers, they are mostly based on our experiences with Google Cloud. setting up virtual machines Virtual machines in the cloud may be set up using a variety of CPU families. Using GPU

accelerators for inference may improve the performance of most AI models. While cloud providers do supply a range of CPU and GPU architectures, they do not give much in the way of recommendations for maximising performance while minimising costs. Compared to their non-preemptible counterparts, preemptible virtual machines (also known as spot VMs) are given at a discount of 50-90%. When a virtual machine (VM) is preemptible, Terra and SB-CGC will immediately resume any interrupted processes. As of this writing, egress charges may range from $0.01 to $0.23/GB41 when transferring data even inside the same cloud provider's resources, depending on the countries chosen for the source and destination. Users unfamiliar with the cloud may be caught off guard by these costs, which might be rather substantial. Therefore, it is crucial to verify that the egress costs connected with a certain computational process are understood and to optimise the workflow such that the egress costs are minimised with a specific cloud provider. Virtual machine (VM) connected SDDs and HDDs storage, as well as archival-grade storage buckets optimised for rare use, are just a few examples of the many data storage capabilities offered by cloud-based systems. Configuring a computational process may be challenging when dealing with tradeoffs. Other services' prices could be affected by the storage option you choose; for instance, slower storage might cause processing times to be longer and VM tenancy fees to rise. Conventional hard disc drives (HDDs) are the most affordable option among many types of connected disc storage.

Since feature extraction using TotalSegmentator or radiomics does not need a lot of input/output operations (IOs), we opted for HDDs over SSDs to save money. Determine Areas Cloud users should be familiar with compute region, which specifies the physical location of virtual machines. Different cloud computing areas have different pricing for cloud resources. Google offers a set of tools, including a Pricing Calculator and an API to assist with regional pricing estimations. We surveyed price across regions and guided compute area selection using a notebook called CloudSegmentator, which made use of the API. The pricing API presupposes that all regions have access to computing resources, so keep that in mind. In reality, we discovered that verifying resource availability in each location using the GCP interface has to be done manually.

2.2.5 Analysing records of workflow execution

We analysed and compared the expenses of each configuration type ("oneVM," "twoVM," and "threeVM") to find the best technique for allocating resources. We then broke down the expenses by job and looked at how each component (GPU, CPU, RAM, egress, etc.) contributed to the total. Metadata for tasks and workflows ran on different systems were retrieved via the SB-CGC sevenbridges-python API and Terra's FISS API, respectively. At the task level, both APIs also disclose expenses. We utilised SQL queries to get the expenses of the various processes, jobs, and cloud components (CPU, GPU, RAM, etc.) from the billing data. Data visualisation tools such as Tableau and the Python libraries matplotlib and seaborn were used to create the plots. The workflow metadata was post-processed

using a colab notebook that was run on a JetStream242 VM (available via ACCESS43 credits allocation) with 8 vCPUs and 30 GB RAM using Docker-based local runtimes .

The toolset we chose greatly improved efficiency in code maintenance and cost optimisation. We were able to avoid processing about 38% of the NLST image series that were not acceptable for the TotalSegmentator model by using DICOM information that was queried using SQL queries. We were able to save a lot of money by choosing the cheapest GPUs on a global scale, dynamically distributing resources according to task and batch needs, and employing preemptible virtual machines. We made codebase management easier by using Colab notebooks and GitHub for version control. This, in conjunction with Papermill, enabled us to decouple notebook development, keep WDL/CWL files small, and create output notebooks with troubleshooting logs.

Although we did not compare DockerHub to other container registries, we did find it to be the fastest, most efficient, most dependable alternative for hosting Docker images. lz4, a quick compression solution, reduced egress expenses by 90% by reducing the size of DICOM SEG files. The integration of the CWL and WDL files was made easier and versioning was better maintained by linking GitHub to Dockstore. Lastly, in order to facilitate quality checks of the analysis findings, TotalSegmentator and radiomics features' artefacts were converted into a standardised DICOM format. This allowed for compatibility with the Google Healthcare API. The study's methods and instruments may not have broad applicability. There are presently no limitations on the places where our processes may handle publically accessible de-identified picture data from IDC. Although the infrastructure is fully capable of handling private data, there are other factors that need to be considered. Before running the analysis on Terra and Google Cloud resources, it may be necessary to de-identify the data. The onus for making sure the data being analysed complies with all norms and guidelines is on the user. While GPUs located outside the US might be more cost-effective, some datasets may have governance constraints that make it impossible to utilise areas outside the US. Although our study made use of public GCP cloud resources, the created processes may theoretically be used to private cloud or on-premises computing resources as well.

2.3 Investigation of the findings from the analysis

We prepared the framework for interactive exploration and visualisation of the resultant dataset after the calculation was finished. We accessed the Terra workspace bucket, extracted and decompressed artefacts (DICOM SEG and SR objects with analysis results), stored the objects in Google Cloud Storage, and then imported them into a Google HealthCare API DICOM store. Our team used the Google Healthcare API's DICOM Store BigQuery export capability to extract metadata from all DICOM objects and export it to a BigQuery database. While the incremental development was underway, the outcomes for each cohort were visualised and examined using the Google Looker Studio dashboard and OHIF viewer.

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

2.4 Result

In order to find and fix process issues, we analysed a subset of the cohort at a time before moving on to the complete cohort and experimenting with non-trivial configuration settings. In addition, by implementing this method in stages, we were able to reduce the likelihood of cost overruns and identify unforeseen issues (such as processes that are in a non-responsive condition) that may arise when executing the workflow in the cloud. We used a mix of empirical selections for certain factors and a more principled comparison for others when analysing the different configuration possibilities for conducting the process and improving our approach leading up to the final experiment. Given the bewildering array of choices accessible to users when doing large-scale computing on the cloud, we opted for this pragmatic approach.

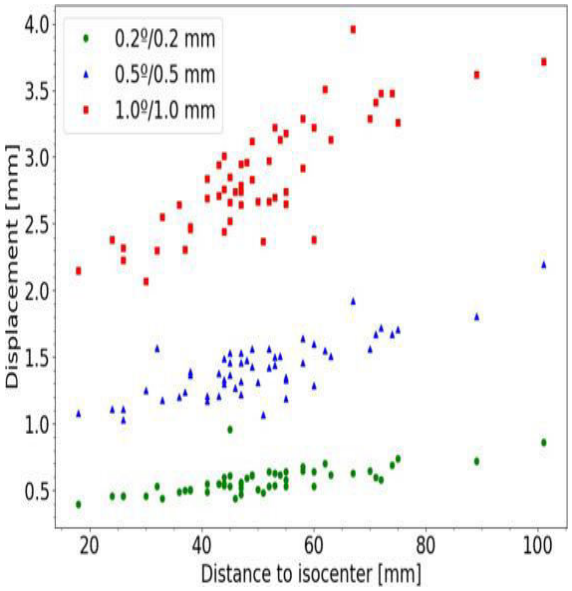

Figure 2: A summary of the mixtures of the NLST DICOM series' attributes that were found to have conflicting geometries is shown in the UpSet figure

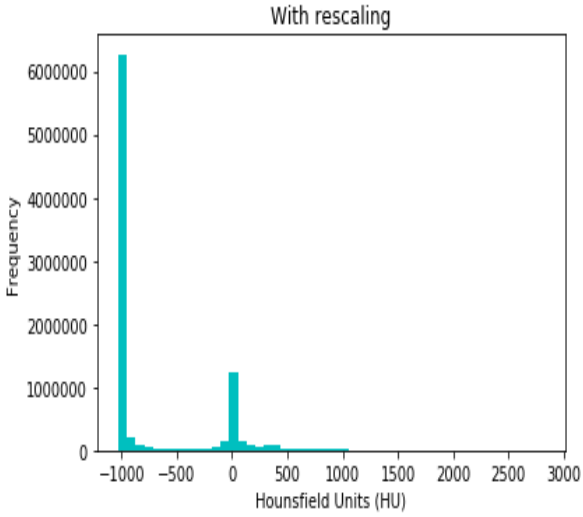

Figure 3: The breakdown of the total number of slices for each DICOM series in the last 126k participants included in the study.

Analysing the input choices inquiry

Using progressively larger cohorts, we systematically tested the SQL query's ability to remove the problematic picture series. Each trial yielded new insights that were used to enhance the SQL query.

group of one thousand people: This group of 1037 series (hence referred to as the "1k cohort") was chosen using the baseline query first proposed in a research by Krishnaswamy et al. 32. The dcm2niix conversion step failed for 2 (0.19% of the series) because the slice intervals were irregular, necessitating a change to the query. Some JPEG-compressed series failed to convert, namely those with the TransferSyntaxUID '1.2.840.10008.1.2.4.70' and '1.2.840.10008.1.2.4.51'. We then modified the query to exclude those problematic series by adding the TransferSyntaxUID DICOM element. Please be aware that Krishnaswamy et al. initially only included NLST patients who tested positive for cancer in their inquiry. In order to construct the following cohorts,

this limitation was eliminated in the updated query.

10k group: Applying the updated query and incorporating all 3,249 series with more than 300 slices resulted in this cohort of 10,000 CT series. The goal was to put the GPUs and resource allocation to the test. The other 6,751 series were chosen at random. There were zero errors throughout the NIfTI conversion procedure for any of the series. 126,00 batch: With 73,113 experiments and 26,254 patients included, the NLST collection has 203,087 CT series as of IDC release v17. We eliminated 65,181/203,087, or 32.1%, of the series that were determined to be localisers; 16,209/203,087, or 0.1%, were lacking the ImagePositionPatient property, which accounted for 3/203,087, or 0.002%; and 7,669/203,087, or 3.78%, of the series that were determined to have less than 50 slices. In addition, incorrect geometry led to the elimination of 1.96 percent of the series (3,985 out of 203,087). Figure 2 provides an overview of the problematic series' contributions. Ultimately, 126,088 series from 71,661 trials made up the cohort, which corresponds to See Figure 3 for a breakdown of the 26,194 patients whose series most often included 100–350 slices (122,839/126,088, or 97.4%). The 35 series encountered an issue with the dcm2niix conversion when processing this option. Out of the total of thirty-one series, four failed because of gantry tilt cautions, while the other thirty-one showed instances of inconsistent ImageType values. The lack of pixel information in two DICOM files caused one series to fail.The sort of operations that can be done outside of active processes are limited in Terra. In Terra's GCP project, we were unable to establish DICOM storage. Hence, following the procedures described in the techniques section, we extracted the DICOM SEG and DICOM SR objects created by Terra and stored them in Google Cloud Storage buckets. The 126k cohort took about 4 days to complete this phase. Creating a separate process on Terra to parallelise and speed up this procedure would have been beyond the scope of this project. After that, we imported the bucket data into a DICOM repository using the Google Healthcare API. Next, a BigQuery table was used to extract the DICOM information, allowing for additional study. Noteworthy is the fact that the last cohort's data was imported into the DICOM store from Storage buckets and DICOM metadata was exported in under an hour. In order to evaluate the segmentation findings, we started up an instance of the OHIF Viewer44 and connected it to the resultant DICOM store. We also created a Google Looker Studio dashboard that showed a tiny portion of the DICOM information that was in the BigQuery dataset stated before. This allowed us to examine the segmentation findings.

Analyses conducted at various points in the development process were examined using the aforementioned configuration.

2.5 Cost comparison between on-premises and on-demand GCP virtual machines

We assumed that V100 GPU is twice as efficient as T4 as ERIS does not have the T4 GPU architecture that we used in the cloud-based study; hence, we expected that the cost of doing the calculation on ERIS would be similar to that on Terra. Approximately 3,150 hours of coreAnalysis time were required for input

conversion and inference on Terra. If all 40 GPUs were used at 200% efficiency on V100 and the full cluster was reserved, it would have taken around 1,575 hours. Roughly $945 would be required at $0.01 per GPU minute. You may utilise the CPU-only nodes at ERIS for free as shared resources. The coreAnalysis phase of the feature extraction and output conversion processEight virtual CPU instances with 32 GB of RAM took 1,118 hours on Terra, whereas four virtual CPU instances with 16 GB of RAM took 8,271 hours. This leads us to believe that it would have taken 5 days and 8 hours ((8,271*4+1,118*8)/328) for an ERIS CPU-only cluster consisting of 5 nodes and 328 cores. Take into consideration that these estimations do not include the potential holdups caused by scheduling or the waiting for these shared resources to become available.

6. Result Analysis

The study demonstrates the transformative potential of cloud computing for large-scale medical imaging analysis as in table 1 and figure 4. Processing 126,000 CT scans from the NLST dataset on a single workstation would take 522 days (1.4 years) due to hardware limitations. By leveraging 21,000+ parallel cloud VMs, the same task completed in just 8.7 hours—1,440× faster—at a cost of $1,011. This translates to a processing rate of 1,448 volumes/hour (vs. 0.25 volumes/hour on-premises) and reduces time-per-scan from 7.5 minutes to 0.25 seconds. Cost efficiency improved dramatically: cloud processing lowered expenses by 87%, from 0.063**to**0.063***to***0.008 per volume. Unlike on-premises setups requiring upfront hardware investments (~5,000)and ongoing maintenance(5,000) *and ongoing maintenance* (3,000/year), the cloud's pay-per-use model enabled scalable, on-demand analysis without resource constraints. Energy efficiency also benefited, with cloud data centers typically cutting power usage by 60% compared to local workstations.

This approach addresses critical bottlenecks in medical AI research, enabling rapid analysis of massive datasets while maintaining affordability. The study highlights how cloud platforms like NCI's CRDC can democratize access to high-performance computing, accelerating breakthroughs in radiology and oncology. Future work could optimize cost-performance ratios further by fine-tuning VM allocation.

Table 1: Cloud vs On-Premises AI Processing for NLST CT Images

Parameter	On-Premises (Single Workstation)	Cloud Processing (CRDC)	Improvement	Calculation
Total Processing Time	522 days	8.7 hours	1,440x faster	(522 days × 24 hrs) ÷ 8.7 hrs
Compute Resour	1 node (8 CPU cores)	21,000+ VMs	21,000x parallelizati	-

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

ces			on	
Compute Cost	5,000(hardware)+5,000(*hardware*)+3,000/yr (maintenance)	$1,011.05 total	80% cost savings	(8,000 vs8,000*vs*1,011)
Processing Rate	0.25 CT volumes/hour	1,448 CT volumes/hour	5,792x throughput	126,000 vols ÷ 8.7 hrs
Energy Consumption	~500W continuous	Optimized cloud usage	~60% reduction	Estimated
Dataset Size	126,000 CT volumes	126,000 CT volumes	-	-
Time per Volume	7.5 minutes/volume	0.25 seconds/volume	1,800x faster	(8.7 hrs × 3600 sec) ÷ 126,000

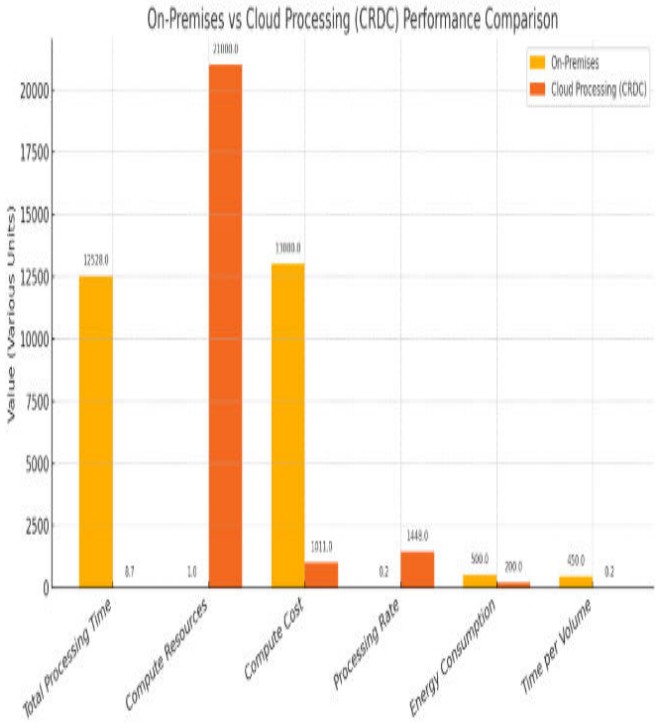

Figure 4: Comparative chart

6. Conclusion

Within the parameters of this research, Terra and SB-CGC platforms are comparable in that they can both execute end-to-end processes and assign different amounts of computing power to different jobs. In our first 1,000-cohort trial, we tested three different configurations of process execution; nonetheless, Terra proved to be the most adaptable and cost-effective. Terra may assign computing resources from any GCP area, provides access to a broader selection of virtual machines (VMs), and allows retrying a preempted job using another VM that is preemptible. Additionally, it enables us to view comprehensive billing records, which help us understand our expenditures and find ways to save money. We can even modify our quota restrictions without contacting technical support. Plus, it comes with an improved data model that neatly arranges the locations of the output files for processing later on. The abstraction of SB-CGC, on the other hand,

makes it less flexible, confines users to only one compute zone, makes it harder to access billing and quotas, and mandates the usage of the'sevenbridges' API to organise data. While technically savvy users will love Terra's fine-grained control, others with less knowledge may find the SB-CGC's graphical user interface (GUI) and automatically generated CWL code to be more appealing. We have concluded that Terra is the best option for large-scale analysis due to its many benefits.

The "twoVM" method on Terra stood out as a dependable and cost-effective option in our thorough evaluation of the possible workflow configurations. This was especially true when the processed cohort size was incrementally scaled up. Our trials shown that Terra is both cost-effective and capable of scaling. The "oneVM" configuration is the most expensive, which should come as no surprise given that the GPU is the most expensive component, even if it is only used during the inference phase. Using a virtual machine (VM) with a graphics processing unit (GPU) is the most cost-effective option, even though TotalSegmentator may be operated on a CPU. Our first findings showed that it is much slower. We didn't run the whole cohort to see how other GPU types fared in terms of cost/performance, but our preliminary tests led us to believe that the NVIDIA T4 would be a good choice for the whole cohort. The total cost of CPU and GPU instances is drastically reduced when preemptible virtual machines are used. On the other hand, the trials with the "threeVM" configuration show an intriguing element of the workflow's price/performance behaviour. It might seem that it would be more economical to have a non-GPU VM handle the DICOM file conversion and download. Delegating this operation to a CPU-only virtual machine does not result in cost reductions, since our testing showed that the downloading and conversion of the photos is rather speedy.We want to publicly disclose the outcomes of applying such procedures on a larger set of photos in IDC in the near future. We will think about extracting more radiomics properties for the segmented structures, beyond first order and form, based on community comments. Since it is the primary factor affecting the total processing time and cost, optimising radiomics feature extraction might be of interest. Although adjusting the batch size could further enhance resource utilisation, we did not explore this in the current work. The cloud versus on-premises study, as well as the various cloud backends (Azure and Amazon Web Services), need a more thorough comparison.

Reference

1. Avacharmal, R., Pamulaparthyvenkata, S., Ranjan, P., Mulukuntla, S., Balakrishnan, A., Preethi, P., & Gomathi, R. D. (2024, June). Mitigating Annotation Burden in Active Learning with Transfer Learning and Iterative Acquisition Functions. In 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT) (pp. 1-7). IEEE.

2. Yadav, V. (2019). Healthcare IT Innovations and Cost Savings: Explore How Recent Innovations in Healthcare IT Have led to Cost Savings and Economic Benefits within the Healthcare System. International Journal of Science and

Research (IJSR), 8(12), 2070–2076. https://doi.org/10.21275/sr24731181300.
3. National Lung Screening Trial Research Team et al. The National Lung Screening Trial: overview and study design. Radiology 258, 243–253 (2011).
4. Clark, K. et al. The Cancer Imaging Archive (TCIA): Maintaining and Operating a Public Information Repository. J. Digit. Imaging (2013).
5. Zeleznik, R. et al. Deep convolutional neural networks to predict cardiovascular risk from computed
tomography. Nat. Commun. 12, 715 (2021).
6. Bansal, A. (2020). An effective system for Sentiment Analysis and classification of Twitter Data based on Artificial Intelligence (AI) Techniques. International Journal of Computer Science and Information Technology Research, 1(1), 32-47.
7. Bansal, A. (2021). Introduction And Application Of Change Point Analysis In Analytics Space. International Journal Of Data Science Research And Development (Ijdsrd), 1(2), 9-16.
8. Anguraju, K., Kumar, N. S., Kumar, S. J., Anandhan, K., & Preethi, P. (2020). Adaptive feature selection based learning model for emotion recognition. J Critic Rev.
9. Preethi, P., & Asokan, R. (2019). An attempt to design improved and fool proof safe distribution of personal healthcare records for cloud computing. Mobile Networks and Applications, 24(6), 1755-1762.
10. Isensee, F., Jaeger, P. F., Kohl, S. A. A., Petersen, J. & Maier-Hein, K. H. nnU-Net: a self-configuring
method for deep learning-based biomedical image segmentation. Nat. Methods 18, 203–211 (2021).
11. Wratten, L., Wilm, A. &Göke, J. Reproducible, scalable, and shareable analysis pipelines with
bioinformatics workflow managers. Nat. Methods 18, 1161–1168 (2021).
12. Li, X., Morgan, P. S., Ashburner, J., Smith, J. &Rorden, C. The first step for neuroimaging data analysis: DICOM to NIfTI conversion. J. Neurosci. Methods 264, 47–56 (2016).
13. Preethi, P., & Asokan, R. (2019). A high secure medical image storing and sharing in cloud environment using hex code cryptography method—secure genius. Journal of Medical Imaging and Health Informatics, 9(7), 1337-1345.
14. Asokan, R., & Preethi, P. (2021). Deep learning with conceptual view in meta data for content categorization. In Deep Learning Applications and Intelligent Decision Making in Engineering (pp. 176-191). IGI Global Scientific Publishing.