Published in final edited form as:
Proc SPIE Int Soc Opt Eng. 2021 February ; 11600: . doi:10.1117/12.2582283.

Web Infrastructure for Data Management, Storage and Computation

Serge Brosset[1], **Maxime Dumont**[1], **Lucia Cevidanes**[1], **Reza Soroushmehr**[1], **Jonas Bianchi**[1], **Marcela Gurgel**[1], **Romain Deleat-Besson**[1], **Celia Le**[1], **Antonio Ruellas**[1], **Marilia Yatabe**[1], **Cauby Chaves Junior**[1], **Liliane Gomes**[1], **Joao Goncalves**[1], **Kayvan Najarian**[1], **Jonathan Gryak**[1], **Martin Styner**[1], **Beatriz Paniagua**[3], **Juan Carlos Prieto**[2]

[1]University of Michigan

[2]University of North Carolina at Chapel Hill

[3]Kitware, Inc.

Abstract

The Data Storage for Computation and Integration (DSCI) proposes management innovations for web-based secure data storage, algorithms deployment, and task execution. Its architecture allows inclusion of plugins for upload, browsing, sharing, and task execution in remote computing grids. Here, we demonstrate the DSCI implementation and the deployment of Image processing tools (TMJSeg), machine learning algorithms (MandSeg, DentalModelSeg), and advanced statistical packages (Multivariate Functional Shape Data Analysis, MFSDA), with data transfer and task execution handled by the clusterpost plug-in. Due to its comprehensive web-based design, local software installation is no longer required. The DSCI aims to enable and maintain a distributed computing and collaboration environment across multi-site clinical centers for the data processing of multisource features such as clinical, biological markers, volumetric images, and 3D surface models, with particular emphasis on analytics for temporomandibular joint osteoarthritis (TMJ OA).

1 Introduction

In the field of medical research, one of the main objectives is to develop tools that can be widely used by dentists, researchers, and the general public. Algorithms for image processing, such as deep neural networks[5,1] and shape statistics [4], are being used more frequently. Such algorithms are at the forefront of healthcare and life science and they are changing diagnostics and treatments. In order to make these algorithms accessible to more users, it is essential to simplify their usability and access to them.

On the other hand, deep learning algorithms require large quantities of samples for training, state of the art performance of deep-learning algorithms is largely driven by the number of samples used for training. Ideally, standardized protocols for data collection in each clinical site should be implemented to increase sample size without compromising data homogeneity, as few training samples or distinct homogeneous data sets might lead to trained models or algorithms that perform poorly when data samples from other sites are analyzed by them. However, implementing such standardized protocols require coordinating efforts among the clinical centers involved.

The Data Storage for Computation and Integration (DSCI) aims to solve these challenging issues by facilitating collaboration across medical centers and offering services for data transfer, sharing, and deployment of algorithms for image processing and statistics. Up to date, the DSCI has particularly focused on research on temporomandibular joint (TMJ) osteoarthritis (OA) and dentistry applications.

In this paper, we describe an open source web-based software solution for data transfer, sharing, and task execution in remote computing grids. We test our framework by deploying algorithms for: 1) automatic image processing tool for TMJ segmentation of small field of view high resolution volumetric images; 2) automatic segmentation of the mandible based on UNET[7] architecture for cone-beam computed tomography (CBCT); 3) automatic segmentation of digital dental models acquired with intraoral scanners, using a modified UNET with residual connections similar to RESNET[2], that we define as RUNET from now on. The algorithms described above produce segmentation of the TMJ condyles and mandibular ramus in both high and lower resolution volumetric images, as well as segmentation of intraoral surface meshes. The graphical user interface (GUI) enables users to perform uploads of large data sets, provides common functionalities for file manipulation and sharing with collaborators, and simplifies the execution of the software tools with the data sets.

The paper is organized as follows. Section 2 describes the materials or data sets used for the experiments. Section 3 describes the architecture of the DSCI framework, and describes the methods or algorithms deployed. Section 4 and 5 shows the results and presents the conclusions of this work.

2 Materials

The de-identified patient data used for task execution, computing, and deploying algorithms in the DSCI are stored in user secure folders, shared only with members of each project. Each project has its specific institutional review board approval. The datasets used int the TMJSeg tool consisted of CBCT scans with small field of view (FOV) that contain only the TMJ region and images of 0.08 mm3 voxel size. The datasets used in the MandSeg tool consist large FOV scans obtained from 3 different centers with voxel size varying from 0.3 mm3 to 0.5mm3. The datasets used int the DentalModelSeg tool consist of digital dental models (DDM) that were acquired with a 3D intraoral scanner that generates 3D models with accuracy of 0.0069 mm.

3 Methods

3.1 DSCI implementation

Figure 2 shows the different components of DSCI. The React[4] library is used to build the graphical user interface (GUI), the Hapi[5] library is used to build the back-end server or orchestrator. Both frameworks allow designing reusable components with scalability in mind. All of the plug-ins are self-contained and are used in other web applications such as CIVILITY[6].

Front-end—The dsci-public serves the web application or static content. The react-hapi-jwt-auth contains the GUI for new account, log in, password reset, user profiles, user scopes (normal user, admin, etc.), and implements the required HTTP requests to handle such transactions. The security and access to the different services and data sets is controlled using Json Web Tokens (JWT) which are encrypted and verified for every transaction. The dsci-filebrowser which is one of the contributions of this work implements a whole set of functionalities for data management (file browsing, copy, paste, move, etc.) and sharing capabilities with collaborators. The clusterpost-list-react[6] is design to manage task execution and monitoring. By combining the file-browser and clusterpost, a GUI component is developed to facilitate task creation and execution.

Back-end—The application is deployed in the Amazon Web Services (AWS) Elastic Compute Cloud (EC2). The Hapi framework, which is designed to facilitate scalability, allows developing plug-ins that are integrated in the application. The couch-provider[7] package is loaded into the Hapi framework and allows other plug-ins/services to discover the functions implementing the operations of couchdb[8] to store, retrieve, delete, etc. entries from the database. The hapi-jwtauth[9] implements the end points to handle all user related request which includes the emission of tokens (JWT) for user authentication and allowing access to other services, and storing user information. The dsci-filebrowser implements the end points for file management and sharing among collaborators. The clusterpost-provider[10] plug-in handles task creation in the server. The task is described by the software/tool to run in the compute node, the data inputs and outputs.

Compute node—The compute node contains containerized software tools. Each container has the specific requirements such as specific versions of the deep learning libraries like tensorflow[11]. The clusterpost-execution daemon checks periodically for new tasks to executes and communicates with the end points in the clusterpost-provider plug-in. Once a task is in the queue, is retrieved and executed. Once the task is completed, the outputs are submitted. The clusterpost-execution is flexible and allows running task in different engines such as UNIX based systems, load sharing facilities (LSF), Sun Grid Engine (PBS), and the SLURM Workload Manager.

3.2 Deployment of algorithms and tools in the DSCI

TMJSeg: Small field of view scan segmentation.—Figure 3 shows the processing steps for the segmentation of small field of view scans. The first step is contrast adjustment

[4] https://reactjs.org/
[5] https://hapi.dev/
[6] https://www.npmjs.com/package/clusterpost-list-react
[7] https://www.npmjs.com/package/couch-provider
[8] https://couchdb.apache.org/
[9] https://www.npmjs.com/package/hapi-jwt-couch
[10] https://www.npmjs.com/package/clusterpost-provider
[11] https://www.tensorflow.org/

of the image by using median and image guided filtering, in order to remove the noise while preserving the shape of the condyle. The contour of the condyle is then detected using a 3D version of the Canny edge detection method. The biggest element by size was removed, which left only the condyle remaining in the scans. That remaining contour was reconstructed to obtain the perfect shape of the condyle. To do so, a convex hull envelope was used on the contour, in order to reconstruct most of the shape. An active contour method was finally used to match the original shape of the condyle and get significant improvement of the segmentation. The code was then package and deployed in our compute node.

MandSeg: Segmentation of large field of view scans.—A deep neural network based on the UNET architecture was trained for this task. The ground truth label maps were manually segmented by clinicians. The UNET takes 2D slices that are extracted from the 3D volumetric images. All the 3D scans were cropped depending on their size in order to keep only the region of interest where the condyle was in the slices. The same anatomic cropping region was used for every scan in the dataset. The scans were acquired at different centers, therefore, every slice is interpolated linearly and resampled to 512×512 pixels. As a preprocessing step, contrast adjustment was performed, because the original scans were low contrast images. This helped the deep learning model to make a better prediction. After image preprocessing, 300–400 slices were extracted from each scan. These were used to train the UNET algorithm. We chose to use a cross-validation method. For that, we take 20% of the data set for testing the models and divide the remaining into 10 folds. The training has been done with 60 epochs, a batch size of 8 and a learning rate of 10^{-5}. So, 10 models were trained using this method, and we obtain the output for the 10 folds. The scans are reconstructed using the slices from the output prediction of each model.

DentalModelSeg: Segmentation of Digital Dental Models.—Fast and accurate segmentation of Digital Dental Models (DDM) remains a challenge due to the various geometrical shapes of teeth, complex tooth arrangements, different dental model qualities, and varying degrees of crowding problems[3]. Figure 4 describes the training approach for the DDM segmentation. A DDM segmentation is produced by acquiring snapshots, running them through the trained model, and setting the resulting label back in the surface. A majority voting scheme is used to decide the resulting value of a vertex if multiple intersections occur.

MFSDA: Multivariate Functional Shape Data Analysis.—The MFSDA is a statistical package that builds associations among features (biological markers, clinical, radiomics etc) and computes the global correlations with morphological variability, as well as local p-values in the 3D TMJ condylar morphology [4].

4 Results

DSCI web user interface

Figure 5 shows different views to perform massive uploads to the system via interactive drag and drop, file browsing to organize and share data sets, and task execution showing the *DentalModelSeg* configuration.

TMJSeg: Small field of view scans.

Results were quantified using the Dice similarity coefficient. The algorithm achieves an average Dice coefficient of 0.95, it is computed using all available small field of view scans. Figure 6 shows a difference between manual segmentation by a clinician (left) and segmentation of the *TMJSeg* algorithm (right).

MandSeg: Segmentation of large field of view scans

Table 1 shows the area under the curve (AUC), F1, sensitivity, specificity, and accuracy measures comparing the ground truth segmentation and prediction using a leave-out-out cross-validation approach.

MFSDA: Multivariate Functional Shape Data Analysis.

Here, the multivariate varying coefficient model tested the association between biological markers and shape morphology. Figure 8 shows the local p-values of association between levels of VE-cadherin in the saliva and the surface mesh of the TMJ condyles in the study sample.

5 Conclusion

We presented a cloud-based solution for computing intensive algorithms including examples of segmentation for small and large field of view scans as well as IOS models. DSCI offers the possibility to the users to launch any algorithm in remote compute nodes. Data transfer and task monitoring are handled by the application. The objective of this framework is to facilitate collaboration of data and algorithms that are containerized and deployed in remote compute nodes. This website can be used for distributed learning storage and management of data collected at different clinics or hospital, and training of algorithms using state of the art neural network architectures. We developed efficient web-based data management, mining, and analytics that integrate and analyze clinical, biological, and high-dimensional imaging data from TMJ OA patients. The Data Storage for Computation and Integration (DSCI) remotely computes machine learning, image analysis, and advanced statistics from patients with and without TMJ-OA. Our long-term goal is to create and maintain the data in a distributed computational environment to allow contributions to the database from multi-clinical centers and to share trained models for TMJ classification.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgements

This study was supported by NIH grants DE R01DE024450.

References

1. de Dumast P, Mirabel C, Paniagua B, Yatabe M, Ruellas A, Tubau N, Styner M, Cevidanes L, Prieto JC: Sva: Shape variation analyzer. In: Medical Imaging 2018: Biomedical Applications in

Molecular, Structural, and Functional Imaging. vol. 10578, p. 105782H. International Society for Optics and Photonics (2018)

2. He K, Zhang X, Ren S, Sun J: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
3. Li Z, Wang H: Interactive tooth separation from dental model using segmentation field. PloS one 11(8), e0161159 (2016) [PubMed: 27532266]
4. Michoud L, Huang C, Yatabe M, Ruellas A, Ioshida M, Paniagua B, Styner M, Gonçalves JR, Bianchi J, Cevidanes L, et al.: A web-based system for statistical shape analysis in temporomandibular joint osteoarthritis. In: Medical Imaging 2019: Biomedical Applications in Molecular, Structural, and Functional Imaging. vol. 10953, p. 109530T. International Society for Optics and Photonics (2019)
5. Nagi R, Aravinda K, Rakesh N, Gupta R, Pal A, Mann AK: Clinical applications and performance of intelligent systems in dental and maxillofacial radiology: A review. Imaging Science in Dentistry 50(2), 81–92 (2020) [PubMed: 32601582]
6. Puechmaille D, Styner M, Prieto JC: Civility: cloud based interactive visualization of tractography brain connectome. In: Medical Imaging 2017: Biomedical Applications in Molecular, Structural, and Functional Imaging. vol. 10137, p. 101370R. International Society for Optics and Photonics (2017)
7. Ronneberger O, Fischer P, Brox T: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)

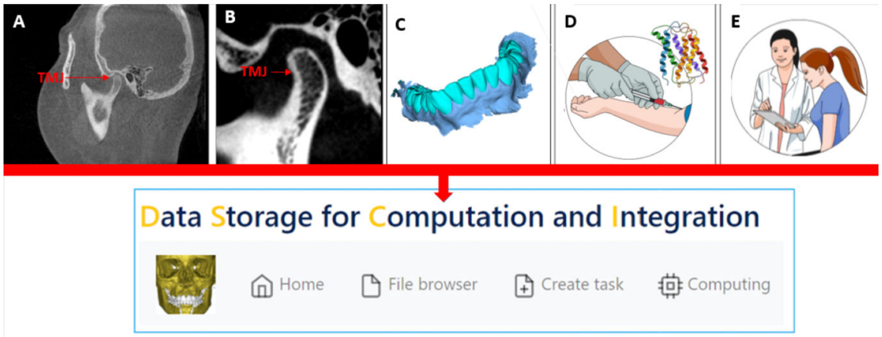


Fig. 1:

Data stored in the DSCI. A, Large field of view CBCT; B, Small field of view CBCT; C, digital dental model; D, biological markers; E, clinical markers

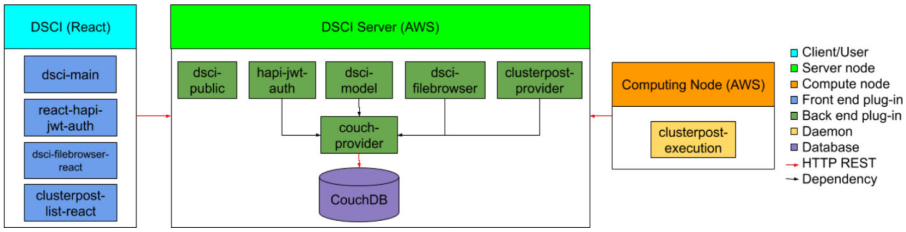


Fig. 2:

DSCI’s architecture. The three main actors of the system: client (cyan), server (green), compute node (orange). The GUI uses the REACT library, the web server uses the Hapi framework, both frameworks allows developing reusable components or plugins. The compute node runs an application or daemon responsible for data transfer to computing grids and task execution.

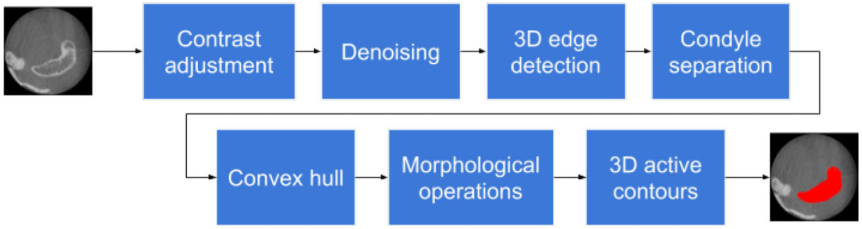


Fig. 3:
TMJSeg processing steps for small field of view scans segmentation.

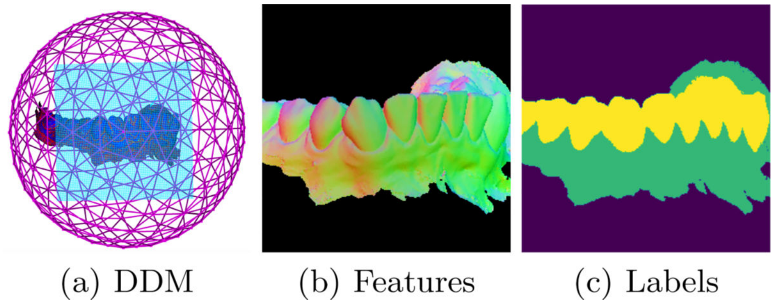
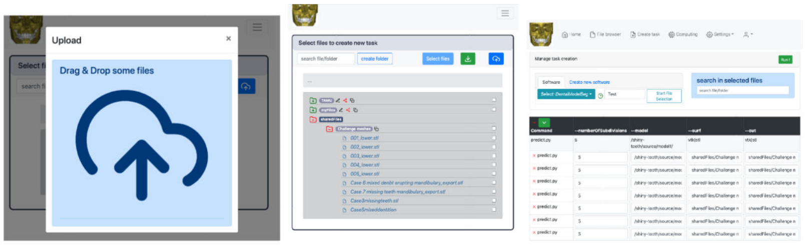


Fig. 4:

The DDM is probed from tangent planes to the unit sphere. 2D images are generated using the distance and normal at the intersection point as features. The images are used to train a RUNET using the corresponding labeled images as ground truth.



- 001_lower.stl
- 002_lower.stl
- 003_lower.stl
- 004_lower.stl
- 005_lower.stl
- Case 6 mixed denbt erupting mandibulary_export.stl
- Case 7 missing teeth mandibulary_export.stl
- Case3missingteeth.stl
- Case5mixeddentition

Command	--numberOfSubdivisions	--model	--surf	--out
predict.py	5	/shiny-tooth/source/modelt/	vtk|stl	vtk|stl
× predict.py	5	/shiny-tooth/source/mo	sharedFiles/Challenge n	sharedFiles/Challenge n
× predict.py	5	/shiny-tooth/source/mo	sharedFiles/Challenge n	sharedFiles/Challenge n
× predict.py	5	/shiny-tooth/source/mo	sharedFiles/Challenge n	sharedFiles/Challenge n
× predict.py	5	/shiny-tooth/source/mo	sharedFiles/Challenge n	sharedFiles/Challenge n
× predict.py	5	/shiny-tooth/source/mo	sharedFiles/Challenge n	sharedFiles/Challenge n
× predict.py	5	/shiny-tooth/source/mo	sharedFiles/Challenge n	sharedFiles/Challenge n
× predict.py	5	/shiny-tooth/source/mo	sharedFiles/Challenge n	sharedFiles/Challenge n

(a) Upload (b) Browse (c) Execute

Fig. 5:

Different views of the React implementation to upload, browse, and execute tasks in the DSCI web application.

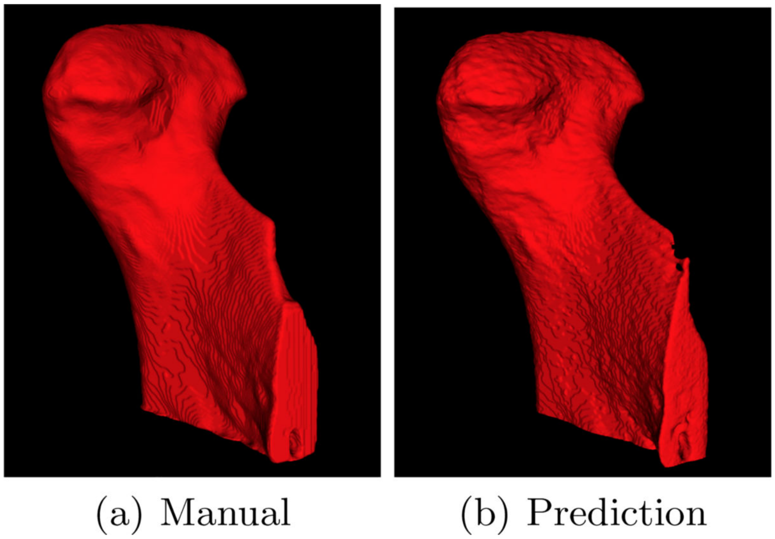


Fig. 6:

Manual ground truth segmentation and prediction of the segmentation for small field of view images. Note how similar the manual and predicted segmentations are.

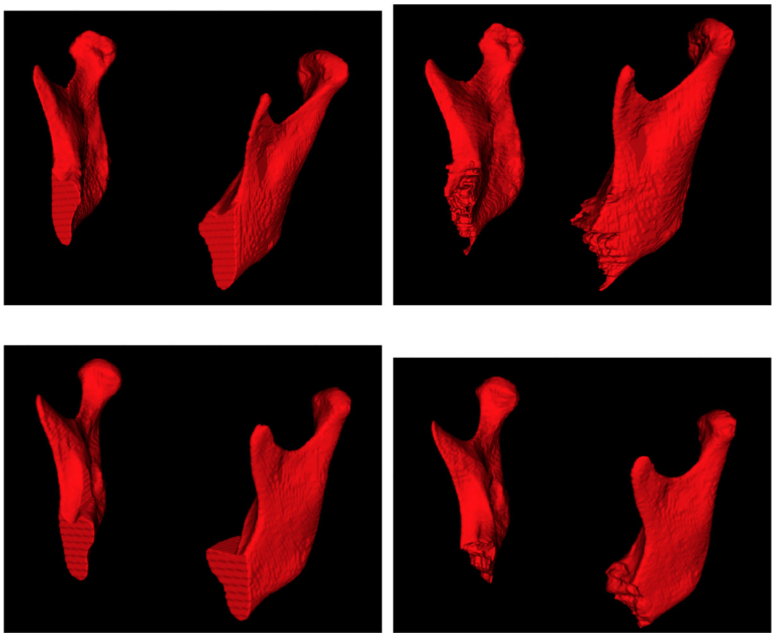


Fig. 7:

Manual segmentations (right), predictions (left), for two different cases.

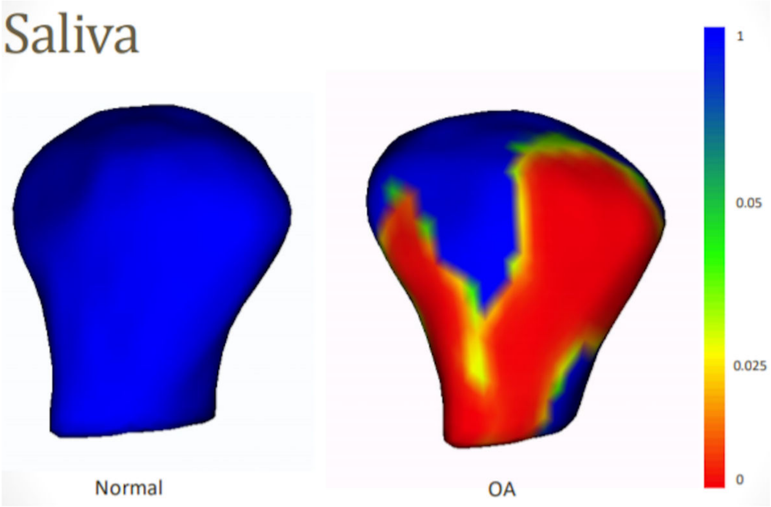


Fig. 8:

MFSDA local p-value maps of the association of Ve-Cadherin levels in saliva and the surface meshes of TMJ condylar morphology

Table 1:

Results for 10 folds using a leave-one-out cross-validation strategy.

N	AUC	F1	Sensitivity	Specificity	Accuracy
1	0.9481	0.9142	0.9207	0.9998	0.9996
2	0.9521	0.9153	0.9248	0.9998	0.9996
3	0.9511	0.9147	0.9228	0.9998	0.9996
4	0.9623	0.9159	0.9383	0.9998	0.9996
5	0.9581	0.9172	0.9335	0.9998	0.9996
6	0.9558	0.9096	0.9265	0.9998	0.9996
7	0.9571	0.9137	0.9322	0.9998	0.9996
8	0.9474	0.9148	0.9201	0.9998	0.9996
9	0.9512	0.9128	0.9243	0.9998	0.9996
10	0.9694	0.9093	0.9443	0.9997	0.9996