

CSC621-821 Biomedical Imaging & Analysis
Fall 2026, Dr. Kazunori Okada
Final Project

Overview: Students will engage in a hands-on group research project for developing and experimenting a biomedical image analysis system using public libraries. You will focus on diagnosing the Chest CT scans of Covid-19 patients to increase our understanding and expertise on this technical and clinical issues. This is a group project consisting of a mixture of undergraduate (621) and graduate (821) students. Each team will *design* and *implement* image data analysis algorithms involving segmentation, registration, and quantification of 3D medical images. Using publicly available dataset, each group then *experiment* the implemented system to gain deeper understanding of 1) how to tune parameters to assure good accuracy, 2) the nature of implemented algorithms, and 3) the nature of the dataset used. Each group will then present their work's outcome *together as a group* at the end of semester, while each student is required to submit a final report created *individually*. Designing aspect of this project plays an important role of your work. **You, the students, will lead the task of deciding the specification of your system and the goal of your experimental studies instead of the instructor!** Student's work will be graded based on the quality and completeness of your individual report and in-class group presentation.

Project Groups: Up to eight teams will be formed. All team members will be assigned by the instructor after the first couple of weeks. Each team should elect a **team lead**. Each team then co-designs and co-manages their own project, following the guideline described below. Basic design of the project should be completed and submitted to the instructor for approval by **9/22**. Team lead(s) are expected to facilitate each group for designing the technical solution, dividing the labor, and planning overall execution.

Final Report: The final report must contain: 1) detailed (long) description of work done by yourself and 2) succinct (short) summary of work done by the entire group. You must provide details of your own individual work in terms of a) relevant technical background info. and technical motivation of your work, b) technical goal/specification of your part, c) technical challenges of the goal, d) the design of your work you conducted, e) implementation aspects of the design, and f) performance evaluation and experimentations of the completed implementation. For the work done by the entire group, you may use up to 2 pages for describing g) technical motivation/reason and goal of the overall system, h) design of the overall system, and i) clear description for individual assignments for all group members. A complete final report should include clear accounts on these nine points (a to i), as well as title, abstract, and reference list.

Final Presentation: The last day of classes (12/8) will be dedicated to the project presentation. Each project team will be given a fixed (**15-30 minutes**) time to present your project work in class. The presentation will be done using an overhead projector. Each team is to have a single presentation slide set shared among all presenters, alternating their turns. Presentation files must be submitted to the instructor a day before

the presentation (12/7 due on 5pm). Live demo is encouraged but not required. A good alternative is to use a pre-recorded video of your system's execution with example data.

Submission: Each student is required to **submit by email a single-spaced 4-to-8 page final report in PDF format** to the instructor (kazokada@sfsu.edu) and your TA Mr. Vinson Thai (vtthai@mail.sfsu.edu). It is **due on Friday 12/11 10pm**. Late policy will apply. Filename of your email attachment should follow the format: "LastName-FirstName-FR621-821.pdf". Subject line of the email should follow the format: "FinalReport-LastName-FirstName". The size of the submission should not exceed 5MB. **Developed codes and their documentations are required to be archived (zip, rar etc) then submitted to the instructor by a project lead.** Presentation files must also be submitted to the instructor a day before the presentation by **Monday 12/7, 5pm**.

Grading: The final project covers **40% of the total grade** for both CSC621 and CSC821. The **individual report covers 20%**. The **group presentation covers 10%**. And the **final 10% is given by peer evaluation of contributions among your teammates**. The report will be evaluated on the basis of 1) evidence of learning, 2) correctness, 3) content's merit, 4) writing clarity, 5) completeness. **Late policy will apply.** A choice of project specification matters for your final grade. More challenging project can be graded higher than the same quality work for easier projects. See below project description for more details. The group presentation will be evaluated on the basis of clarity in organization/delivery and correctness in content. **Be concise, punctual, while emphasizing the main findings of your study. Each member of a group will have the same grade.** A *Canvas* survey will be used to collect your opinion on how each team member contributed to your project at the end of semester. Average percentage of your contributions judged by your peers will be used for your grade.

Coding Platform: Students should use an API called Insight Segmentation and Registration Toolkit (ITK <http://www.itk.org>) with C++ or (SimpleITK <https://simpleitk.org/TUTORIAL>) with Python, which provides most of necessary algorithmic components to build your final system. ITK is a freeware which can be downloaded from the above site. **It is student's responsibility to familiarize themselves with ITK early in semester.** There are numerous online tutorials which can be easily found online. For example see <http://www.itk.org/ITK/help/tutorials.html>, <https://github.com/KitwareMedical/CourseInBiomedicalImageAnalysisVisualizationAndArtificialIntelligence>, and <http://www.itk.org/ItkSoftwareGuide.pdf>. ITK also uses CMake and can run on various platforms, including Windows, Linux and Mac. Each group is suggested to pick a single OS to be shared among all members. Use a virtual machine solution if needed. IDE can be chosen freely but Netbeans has not worked well with CMake in past. MS Visual Studio should work fine. Members of the same group should also use the same IDE whatever the choice will be. For building GUI for your application, the common choice includes VTK (<http://www.vtk.org/>) and FLTK (<http://www.fltk.org/>). **It is also student's responsibility to familiarize themselves with these GUI tools early in semester.** For code version management, each group is allowed to use any tools, such as GitHub, following the university and departmental plagiarism policy strictly.

Discussion Forum: Students are encouraged to participate in online discussions for learning toward your final projects. *Canvas* forum will be set up for this purpose. (Note. *Canvas* accounts for CSC621 and CSC821 will be combined and will be called as CS621 page to be used by all of us). **Be aware that the learning curve of ITK, VTK, and FLTK APIs are steep. Start early to make your hands dirty.** Ask questions in the *Canvas* forum as well as help others by answering other's questions. Active involvement in discussions will be favorably counted toward your final grade.

Data: There are various biomedical data available online. In this project, I encourage you to focus on 3D (volumetric) medical images/scans. The Cancer Imaging Archive (TCIA) <https://www.cancerimagingarchive.net/collections/> is one of the most comprehensive public image data archives organized as a part of the National Cancer Institute of the National Institute of Health. They offer various anonymized clinical images that sometime come with expert annotations. Downloading these image datasets will require you to install the NBIA Data Retriever. Information of this can be found here: <https://wiki.cancerimagingarchive.net/display/NBIA/Downloading+TCIA+Images>. There are many other pools of image data and pointers that are organized by research organizations and/or crowd sourcing efforts. ~~For example, see Stanford's archive at: <https://aimi.stanford.edu/research/public-datasets>~~ You can use any image sets from these or any other archives publicly available as long as they are anonymized to protect the privacy of patients. To view these volumetric data, you must also download DICOM and other medical image specific image viewer, such as **ImageJ**: <https://imagej.net/ij/index.html> and <https://imagej.net/ImageJ>. When accessing compressed DICOM images that cannot be read by native ImageJ, you can install bio-formats plugin from <https://www.openmicroscopy.org/bio-formats/>. Follow instruction in <https://docs.openmicroscopy.org/bio-formats/6.6.0/users/imagej/installing.html> to install and use the plugin in ImageJ. ITK also comes with image I/O classes that can load these image files into memory for your project work.

Research Topics: Each team is to build an image diagnostic system that analyzes chest CT scans of Covid-19 patients. Download a recent Covid-19 image data share of TCIA at <https://wiki.cancerimagingarchive.net/pages/viewpage.action?pageId=80969742>. This dataset includes 120 CT scans with expert's data annotations in JSON format. For more details of this dataset, read: <https://pubs.rsna.org/doi/10.1148/radiol.2021203957>. There are additional data sources you may explore at your discretion. For example, see: <https://wiki.cancerimagingarchive.net/display/Public/CT+Images+in+COVID-19>.

For designing your system specification, use the following guidelines. Your final system should include three main components that performs **segmentation**, **registration**, and **quantification**. It does not require a dedicated standalone GUI that unifies all components. Each component can save the results into specific format files then you can use any existing viewers such as ImageJ/ITK-SNAP or simple viewer built by you using VTK. Segmentation can be applied to any chest scans to delineate the area of lung fields or area infected by Covid-19. Registration can be applied to a pair of scans to align them spatially so that the patient's bodies or lung fields overlap to each other, enabling a

quantitative comparison. Quantification include a) experimenting different values of system's parameters to find the best tuning, b) validating your segmentation results by comparing your results against JSON annotations, c) validating your registration results by checking the alignments visually/quantitatively, d) compare different segmentation/registrations algorithms to determine which one perform best for the used data, e) evaluating changes of shapes/size/nature of infection across different patients, f) detecting a case of Covid-19 patient, and g) quantifying severity of Covid-19 infections.

In planning your team's project, first, choose one or a few algorithms that you will implement for segmentation and registration, respectively. Second, choose three or more types of quantification tasks listed above for you to perform. Divide the team into two sub-teams first and assign either segmentation or registration system development. Each sub-team will be responsible for respective component's implementation and testing. Parallel to this division, also divide your team into three or more groups and assign the chosen quantification tasks. Create a set of milestones for these implementation and experimentation tasks and check in on/help each other by meeting weekly at least. Note that certain aspect of quantification requires a completed system therefore you want to put milestones for quantification tasks later than those for segmentation/registration implementation. Toward the end of the term, also allocate the time for all team members to work together to integrate all the results and prepare for the project presentation.

Details of segmentation, registration, and quantification of medical images will be taught in this course in depth, but they will come at the end of the semester. To help you plan your project specifications before you learn these materials in class, I will give some brief overviews and resource pointers below:

You are encouraged to use publicly available ITK libraries and codes (including the sample ITK solutions that can be downloaded from ITK archive) to implement two major components of your system: A) Registration-ChangeAnalysis and B) Segmentation-Quantification. Although more challenging designs, tasks, and self-implementation will be rewarded with higher grades, each group should carefully design your solution so that you can complete your work within the short time given.

(A) Registration-ChangeAnalysis component asks you to build a subsystem to automatically detect and monitor abnormal changes of a patient over time. Your system takes a pair of 3D volumetric scans as input. You will design and implement the successive processes of i) performing a pair-wise registration to align the pair of volumes geometrically by choosing one as a reference then transforming the other to match the reference, ii) visualizing the significant change between the aligned volumetric image pair, and iii) analyzing the change between the pair quantitatively. The entire procedure can be designed with manual user-interactions (involving a GUI) or automatically (more difficult). The following four algorithms are suggested for this task as available in ITK:

- A1. Landmark-based rigid registration with Iterative Closest Point algorithm with manual selection of correspondences using GUI**
- A2. Intensity-based rigid registration with mutual information maximization**

A3. Non-rigid registration with B-Spline deformable transformation

A4. Non-rigid registration with Thirion's demons algorithm

Mathematical morphology operator is useful for separating closely located objects that are accidentally recognized as a single object. Connected component labeling is useful for counting isolated regions. Refer to the following ITK tutorials for reference:

<https://bidal.sfsu.edu/~kazokada/csc621-821/RegistrationMethodsOverview.pdf> and <https://bidal.sfsu.edu/~kazokada/csc621-821/NonRigidRegistrationMethods.pdf>.

(B) Segmentation-Quantification component asks you to build a subsystem for screening a patient's vital information automatically. In this project, you target clinical information about Covid-19 infection in patient's lungs. Your system takes a 3D volumetric scan as input. You will design and implement the processes of i) delineating 3D regions belonging to lung fields from an input 3D volumetric scan, ii) delineating 3D regions showing infections by Covid-19, and iii) characterizing quantitative information of the infection (e.g., size, shape, texture) of each segmented region. Given segmentation results, you can evaluate and compare the performance of the implemented system(s) by comparing the results against expert annotated segmentation in JSON. The entire procedure can be designed with manual user-interactions (involving a GUI) or automatically (more difficult). The following four algorithms are suggested for this task:

B1. 3D edge-preserving smoothing followed by a threshold-based segmentation technique

B2. 3D Watersheds segmentation and fusion method

B3. 3D Region growing segmentation method

B4. 3D deformable model-based segmentation using Level Set method

Mathematical morphology operator is useful for separating closely-located objects that are accidentally recognized as a single object. Connected component labeling is useful for counting isolated regions. Refer to the following ITK tutorial for reference:

<https://bidal.sfsu.edu/~kazokada/csc621-821/SegmentationMethodsOverview.pdf>.

(C) Quantification for parameter tuning and performance validation can be evaluated qualitatively by inspecting exemplar case results visually and quantitatively by using standard statistical figure-of-merit measures (e.g., Dice Coefficient, F-measure etc) for segmentation and image difference for registration. For change analysis over time or across patients, standard statistical analysis of variances can be applied to any set of values computed. Detection, classification, and severity quantification of Covid-19 infections in Chest CT scans are more advanced topics and this course will not cover them in the lectures. Ambitious teams are still encouraged to try any of them by studying the related literature to try them. Convolutional neural network (CNN)-based solutions have been shown to be effective to these tasks and there are many python-based libraries available.

More examples of ITK applications can be found under example folder of the ITK sources as well as at <https://blog.kitware.com/courses-in-medical-image-analysis-that-use-itk/>