

Medical Imaging Coding Assessment

Objective:

Develop a web-based platform where users can upload a DICOM image. Upon upload, the pixel data of the DICOM should be extracted, normalized, and thresholded. After processing, the system should display the volume (in mm^3) of the thresholded pixels.

Requirements:

1. Web Interface:

- A simple web page with a drop zone for uploading DICOM images.
- Display the number of pixels that are above the threshold once the processing is complete.
- Although we do not want you to spend that much time on the UI elements, it is important that the interface is functioning as intended.
- We recommend Vue.js, but feel free to use whatever you are familiar with.

2. Backend:

- REST Endpoints for communicating with the Web interface. All the endpoints should return JSON data.
- We recommend using the FastAPI Python framework, but feel free to use any other language/framework.
- Extraction of pixel data from the DICOM (We recommend the Python package `pydicom`).
- Normalize the pixel data to a range of $[0, 1]$.
- The pixel data should be thresholded($>$). Use the threshold value specified in the web server's config file (default to 0.5 if not provided).
- Return the volume of the thresholded pixels in mm^3

3. Docker:

- Package the application in Docker images.
- Provide `Dockerfiles` for both backend and frontend
- Ideally, a `docker-compose.yml` for running the solution. If you prefer an alternate approach (such as bash scripts), ensure that the `README.md` file clearly explains how we can test your application.

4. Code quality and testing:

- Your code should be clearly understandable, well-documented and partially tested.
- Think of edge cases which might occur. You don't have to cover all of them, but ideally you are aware of them and would know how to solve them given more time.
- For testing use the DICOM 1-101.dcm from this folder, where the expected volume is $\sim 143,280.029 \text{ mm}^3$.

5. Future work

- Think of the limitations with the current implementation.
- What would you improve in the future?

Submission:

1. A GitHub repository or a .zip containing all your code.
2. A `README.md` with instructions on how to run the application, including building and running the Docker containers.

Good luck with your coding assessment, and we look forward to reviewing your solution!

If you have any questions, please don't hesitate to contact us.