Robotic Process Automation in a Remote Instrumentation and Collaboration Environment (RICE)

-- Image Analytics Platform as a Service

Mentors : Mauro Lemus, Songjie Wang

Team Members : Patrick Wisnewski, Preyea Regmi, Divya Mishra, Abdullah Maruf, Thadeus Meneses, Parshad Suthar, Austin Neumann, Caroline Lewis.

Cloud Computing I Spring 2022

Table of Contents

1. Project Scope and Overview

With the advent of cloud computing and the vast array of services that they offer, every domain is trying to leverage cloud in one way or the other. The scientific community is also not an exception, every modern scientific instrument supports integration with cloud for seamless experiments and collaboration. However, the old expensive instruments that predate the cloud are also still being used today. These instruments are controlled by domain specific software which has no direct compatibility with external software/services and always requires human intervention to operate.

This project aims to mitigate the human intervention required in such scientific instruments by incorporating AI-based control feedback mechanisms and make them autonomous for performing remote instrumentation.

2. Methodology And Objectives

Scientific instruments are built to carry out specific experiments that are critical to their domain. One such example is Scanning Electronic Microscope (SEM) that is used for biological image segmentation of Mitochondria. This type of experiment is centric toward the analysis of the captured image and the instruments that are used need to be calibrated to achieve good results. A typical workflow for these devices consists of a human operator and a SEM Controller Interface that is used to control these SEM devices. Each image captured through these devices needs to be evaluated for correctness before they can be claimed as a fact for the experiment.

The proposed solution relies on cloud services and incorporation of AI/ML techniques to automate the image data collection/transfer/analytics. The accumulated knowledge base from this collected data helps to formulate the AI-based control feedback mechanism to remotely control and calibrate the SEM devices.

3. Software

The software team worked on three different components required for the project using the mentioned technologies

I. Sem Controller
   Phenom Programming Interface and Python Flask.
II. Sem Intelligent Agent
   Image processing using Python, Amazon S3, Local Storage and API Development using Node.js
III. Frontend and User API
   Angular 12 and Node.js

a. Core Logic

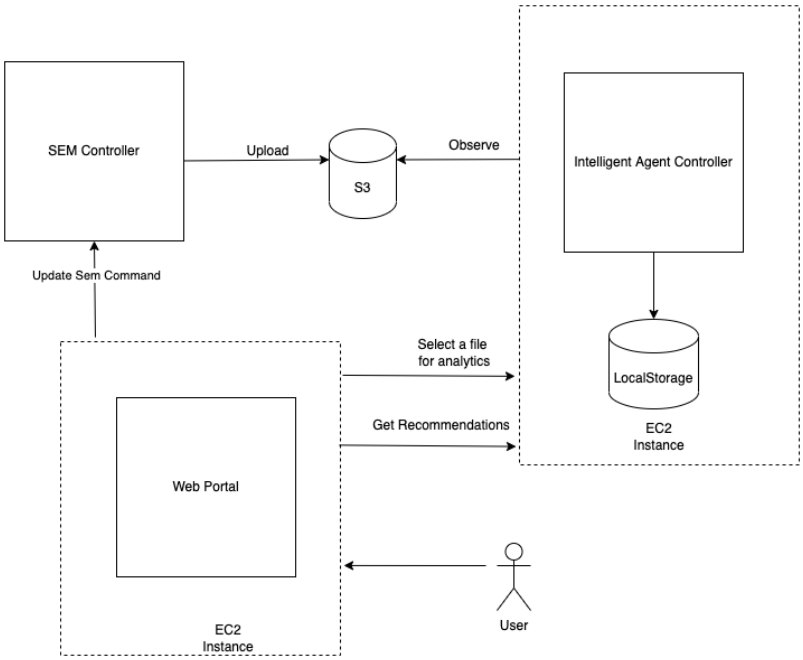

Figure 1 End-to-End Interaction Diagram

Our system consists mainly of three main components which are Sem Controller, Sem Intelligent Agent, and a Web portal through which remote instrumentation can be carried out. The SEM controller interfaces with the actual microscope and can be controlled remotely. Then the images captured by the microscope are uploaded to Amazon S3 through the SEM controller.

Users/Researcher can login through a web portal where all the collected images along with their attributes are listed. The images which are yet to be evaluated can be sent toward the Intelligent Agent Controller for image processing. The Intelligent Agent Controller observes the images that were uploaded by the SEM Controller. When a user selects a file for analytics, the image is downloaded to local storage and further evaluated to obtain the dice score. Based on dice score, if any reconfiguration is required then corresponding recommendation actions are prompted to the user and send SEM command to the Controller.

b. Functional Requirement

The functional requirements of this project include:

- User interface development for managing the services and user access of the services remotely.
- REST API development for the integration of the services.
- SEM Agent development.
- Train ML/DL model to implement control feedback mechanism.

- Deploy on cloud and expose Software-As-A Service for Analytics and Data Repository.

c. Non-Functional Requirement

The non-functional requirements of this project include:

- Literature review of similar projects.
- Identify, implement, and deploy cloud testbed requirements.
- Performance evaluation using collected data.
- Perform integration testing of developed services.

d. Image Segmentation and Analysis

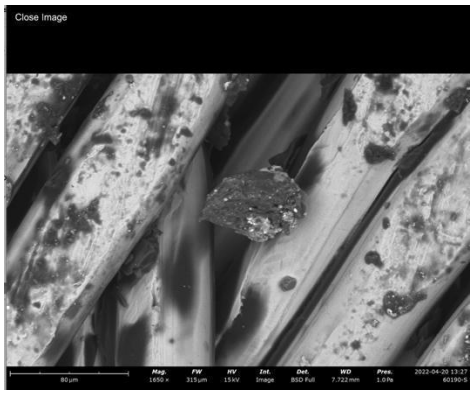

Figure 2 Sample image captured by SEM

```
/home/ubuntu/.local/lib/python3.8/site-packages/torch/nn/functional.py:1944: UserWarning: nn.functional.sigmoid is deprecated. Use torch.sigmoid instead.
  warnings.warn("nn.functional.sigmoid is deprecated. Use torch.sigmoid instead.")
Batch 1/1
====================Semenatic Segmentation====================
Rat eval volume 0, Dice score = 0.5684051513671875


====================Instance Segmentation====================
---------Volume 0---------
        2. Compute IoU
         compute bounding boxes
100%|
         compute iou matching
4it [00:00, 1863.31it/s]
        -RUNTIME:       0.172 [sec]

start evaluation
Accumulating evaluation results...
```

Figure 3 Dice Score Evaluation of Image

Upon user request, the collected images from SEM Controller are used for image processing and corresponding dice score is obtained. The evaluated dice score helps to determine the quality of captured images without human intervention. We have currently used a threshold of value 0.7 as quality margin denoting the correctness of captured images. If the value is less than 0.7, we recommend users to take some action to reconfigure the SEM. For instance, in the above evaluation we obtain a dice score of 0.56 due to low quality of image. So “Change Resolution” action is recommended to the user.

4. Image Processing Test Result

The idea is when an analysis is requested on an image, the intelligent agent controller will pull off the image from the s3 storage, then run a background process which will run the analysis on the image. As the image analysis will take a good amount of time to complete it will run asynchronously and will store the result in the database. While running the image analysis, the intelligent agent will generate a unique id for each of the processes and the result of that process will be available when the process is done with the analysis. Then the workflow will be analyzing the result of the image processing score known as dice score, researchers can invoke another analysis on the same image again. Also, from the front end researchers can send new commands to the SEM controller with new params and pull new images for analysis. or researcher can also set a params change on the score and then automate it. For instance, if the score is like this then set the params + or - then send the instruction to the SEM controller to pull the image again. and when done run the analysis again, generate the analysis result and check with the condition. and run the process again until the condition is met. Right now, only the researcher can send control commands to the SEM. The automation is not working as the SEM controller API is not available now. The whole workflow is illustrated as below.

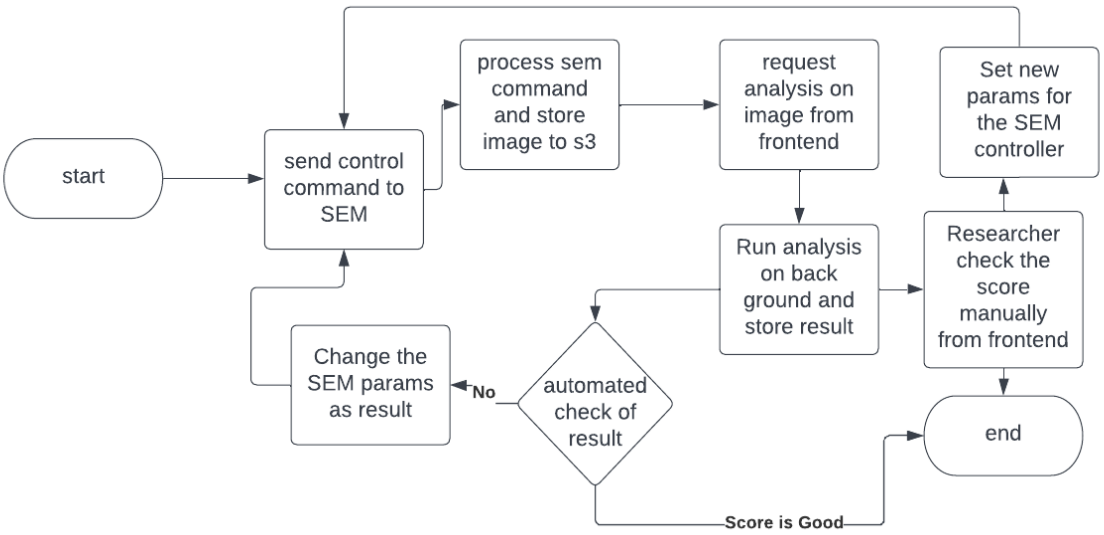

Figure 4 Image Processing Workflow

5. API and Use case

The individual components are integrated using RESTful API. An API document was created prior to starting the development for collaboration. This allowed us to get an early overview of the overall system and as well as to determine the edge cases.

The following diagram illustrates the use-case of individual components.

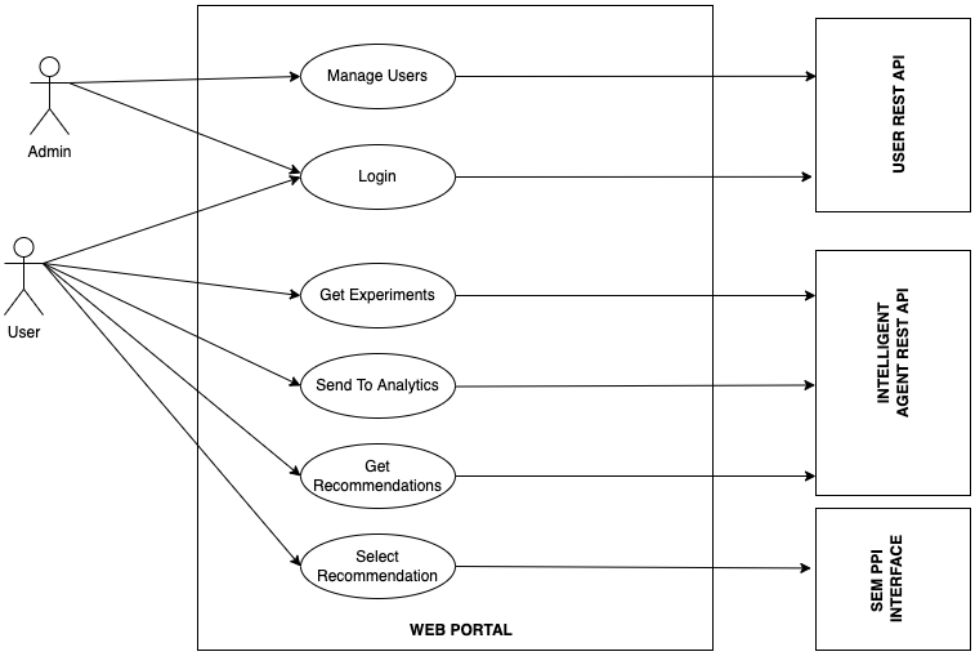

Figure 5 Web Portal Use Case

We have applied two roles that a user can take i.e., either 'Admin' or 'User'. Admin are assigned with an additional functionality to manage rest of the users while other functionality remains the same. Login is an entry point to our application. After successful login, users can view all the images that were captured by the microscope through the Intelligent Agent REST API. Currently there are issues regarding the remote connection to the SEM instrument. Initially these issues were caused by firewall issues and network protocol issues. We are currently working on how to resolve these issues while maintaining the firewall settings. Once they are resolved the API will operate as planned. For the images that are yet to be evaluated, user can send to analytics for image processing and based on the obtained dice score fetch corresponding recommendation. The recommendations are a mapping of actions and corresponding SEM commands which SEM controller understands.

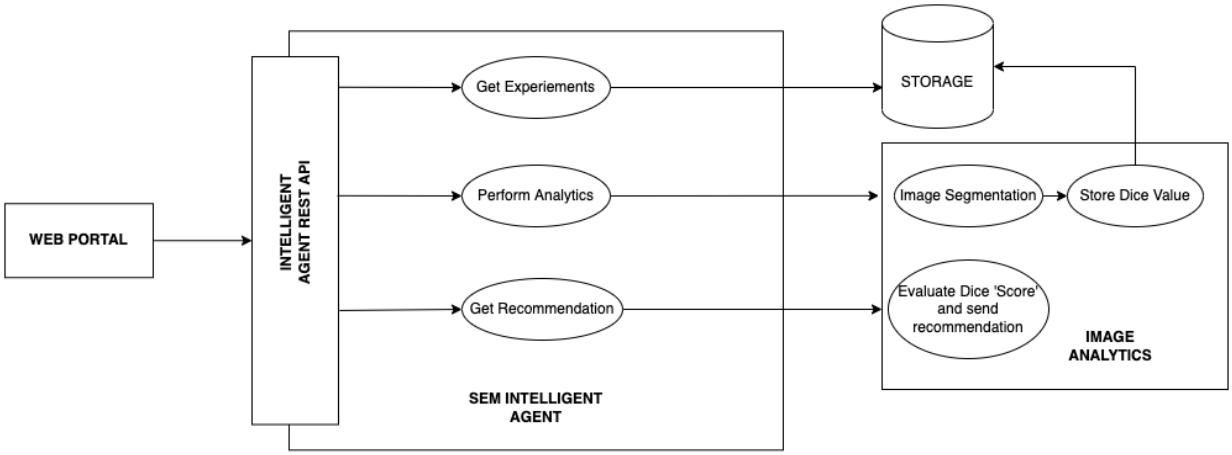

Figure 6 Intelligent Agent Use Case

The SEM Intelligent exposes API to the web portal as described earlier. During the image analytics phase, image segmentation is performed on the corresponding dice score and is persisted for later use. Users can make full corresponding recommendations based on the dice score obtained.

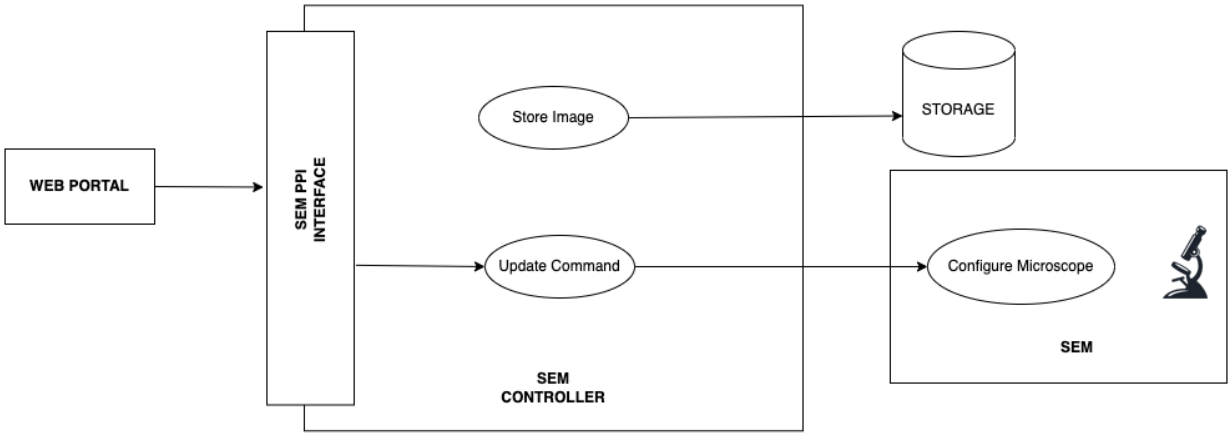

Figure 7 SEM Controller Use Case

The SEM PPI also exposes the API to receive the SEM command from the web portal. It also persists every image that is captured by the microscope to S3 storage.

6. Testbed Setup

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
RiceWebServer	i-007692048ce251f8b	Running	t2.micro	2/2 checks passed	No alarms	us-east-2b	ec2-18-118-216
CMLServerLarge	i-0d78f60e86f73de75	Running	t2.large	2/2 checks passed	No alarms	us-east-2a	ec2-3-138-106-
SEMController...	i-024bcbbbb48880bdf	Stopped	t2.micro	–	No alarms	us-east-2a	–
SEMAPIContro...	i-08803db24949b38d1	Stopped	t2.micro	–	No alarms	us-east-2a	–
SEMController...	i-0fc3af983dd11c92f	Running	t2.micro	2/2 checks passed	No alarms	us-east-2a	ec2-18-119-107

EC2 Instance 1 – RiceWebServer
Instance Type : t2.micro
RAM : 1GB
vCPUS : 1
WebServer : NGINX

EC2 instance 2: CML
Instance Type : t2.large
RAM : 8 GB
vCPUS : 4
WebServer : nodejs
Analyzer: Written in python with pytorch; invoked as a separate process to run analysis by server.

EC2 Instance 3 – SEMController:

Instance Type : t2.micro
RAM : 1GB
vCPUS : 1
WebServer : IIS (Windows 2019 Server)

http://ec2-18-119-107-254.us-east-2.compute.amazonaws.com/

- APIs can be accessed via EC2 instance that is a Python Flask Server
- Allows you to interact with SEM via PPI through the Web Portal
- For further reference on how the PPI works see the PPI User Guide here: https://drive.google.com/file/d/1eRQQGZCkNIikhXDSDpirQ6HFTc4PSIED/view?usp=sharing.
- For a stronger understanding of how the API and endpoints work, see this Postman file you can import that allows you to test particular endpoints and values: https://drive.google.com/file/d/1CoKDqcqYX8JKbYFMPzsSPk6v3MzxOD-6/view?usp=sharing.
- Also you can see the actual server file itself for deeper understanding of the endpoints: https://drive.google.com/file/d/1hyIS1CKUZQ3HRIzOWlXBuZcnEtM58vYe/view?usp=sharing.

Integration Interface	Dataset	Mode of Communication (aka Access/Transfer Method)	Storage Management
1. From Storage Service to Web Portal	**Image information:** Image metadata, scan parameters, analytics results, shared status **SEM Image:** Image raw information for display purposes	**REST API.** The web portal retrieves the metadata and the images.	Images and metadata are handled by the Python code within data structures
2. From Web Portal to SEM PPI interface	Image information, SEM image	PPI	PPI/Cloud Storage
3. From SEM PPI interface to SEM	Image information, SEM image	Python Flask Server	Local storage
4. From SEM PPI interface to Storage Service	**SEM Images:** Images collected by the microscope upon request by executing the scan image command	**REST API. SEM API:** Images are retrieved using the PPI data load command	**AWS-S3:** Images are stored as xxx.xxx files within a S3 bucket using AWS CLI commands
5. From Storage Service to Intelligent Agent	**SEM Images:** New Images arriving to the storage service are loaded to the IA for analysis	The IA uses AWS CLI commands to retrieve the images tagged as NEW	Images are managed by the Python code within an array
Integration Interface	**Dataset**	Mode of Communication (aka Access/Transfer Method)	Storage Management

1. From Web Portal to Intelligent Agent	**Processed Image Data**	REST API	Stored via Storage Service
2. From Web Portal to SEM PPI interface	**SEM Commands**	REST API	AWS Cloud Storage
3. From SEM to SEM PPI interface	**SEM Commands, SEM Images, Image information**	PPI	Local computer
4. From Storage Service to SEM PPI interface	**N/A**	N/A	N/A
5. From Intelligent Agent to Storage Service	**IA image analysis result: the result from the image analysis process are sent back to the storage service**	Results are stored in a DB using regular SQL commands	Results will be stored within a particular table with certain parameters

7. Result

http://ec2-18-118-216-31.us-east-2.compute.amazonaws.com/login

Username: admin / user1
Password: admin / 123456

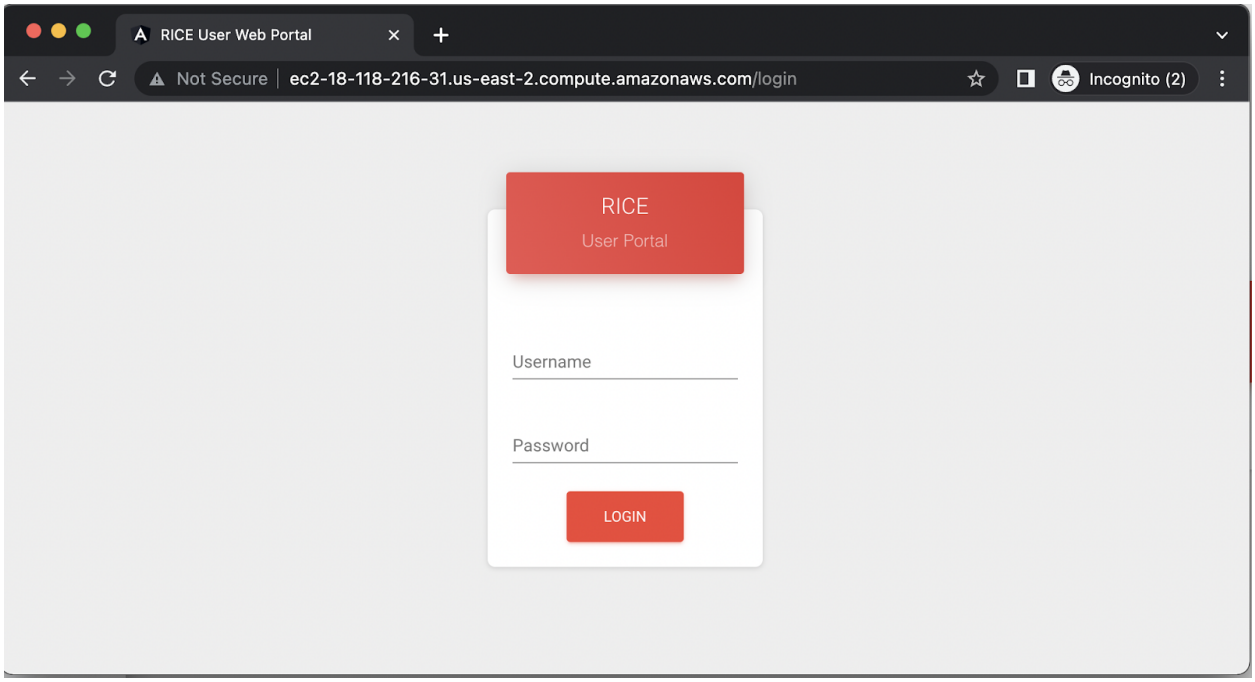

Figure 8 Login Portal

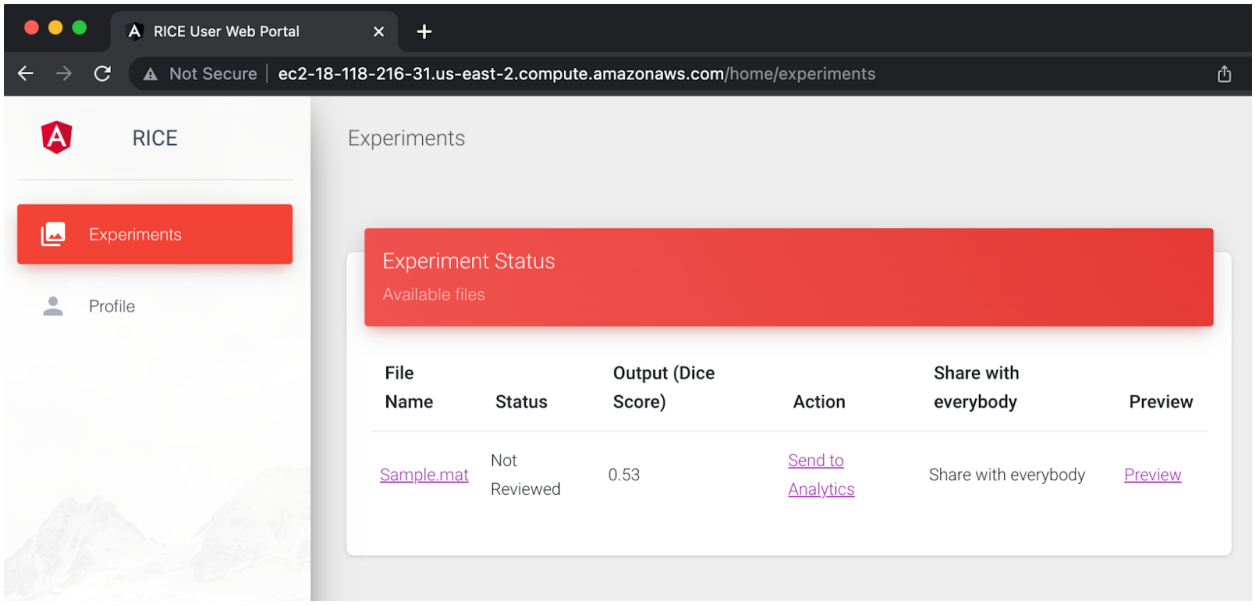

Figure 9 Experiment Listing Screen

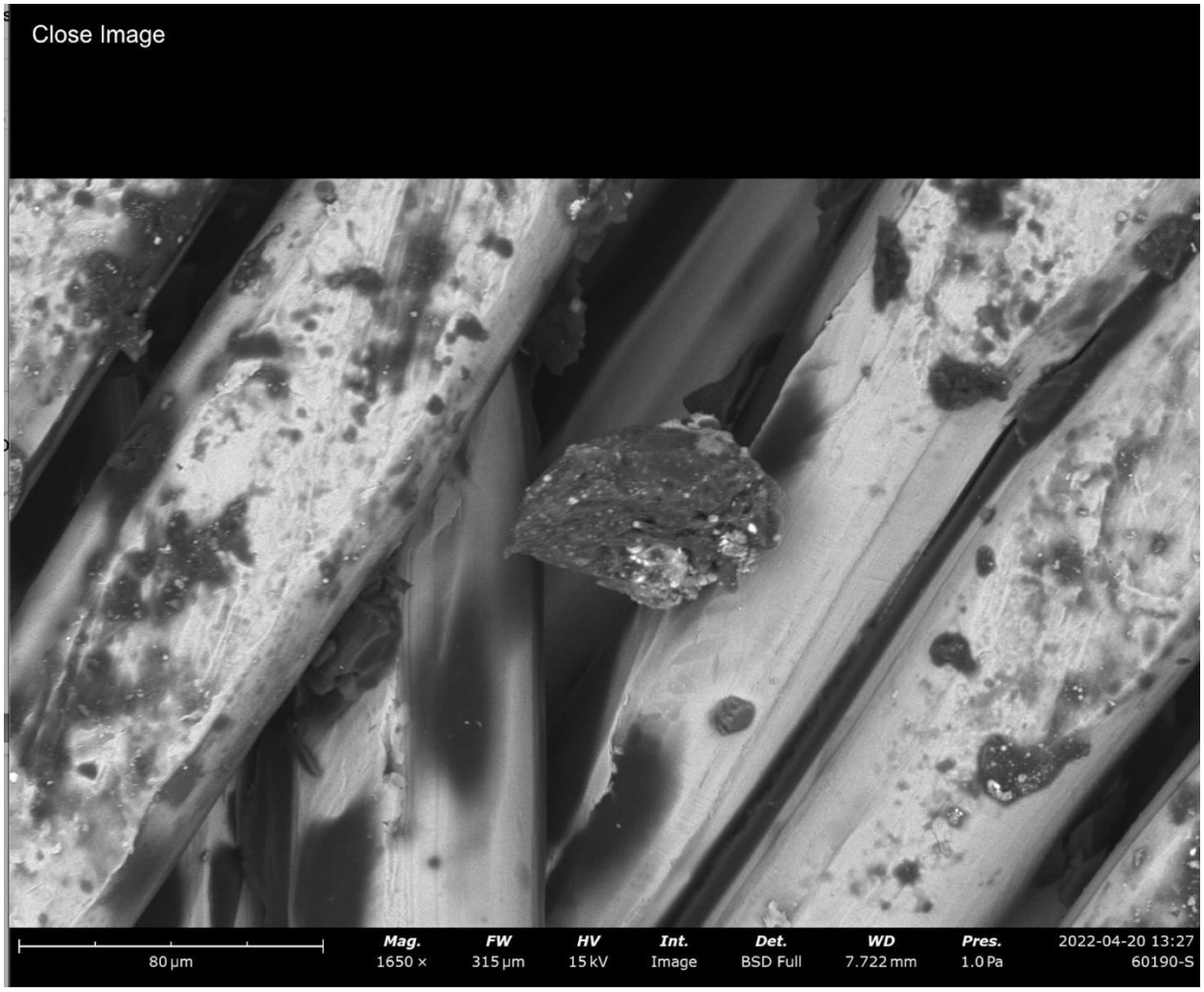

Figure 10 Image Preview

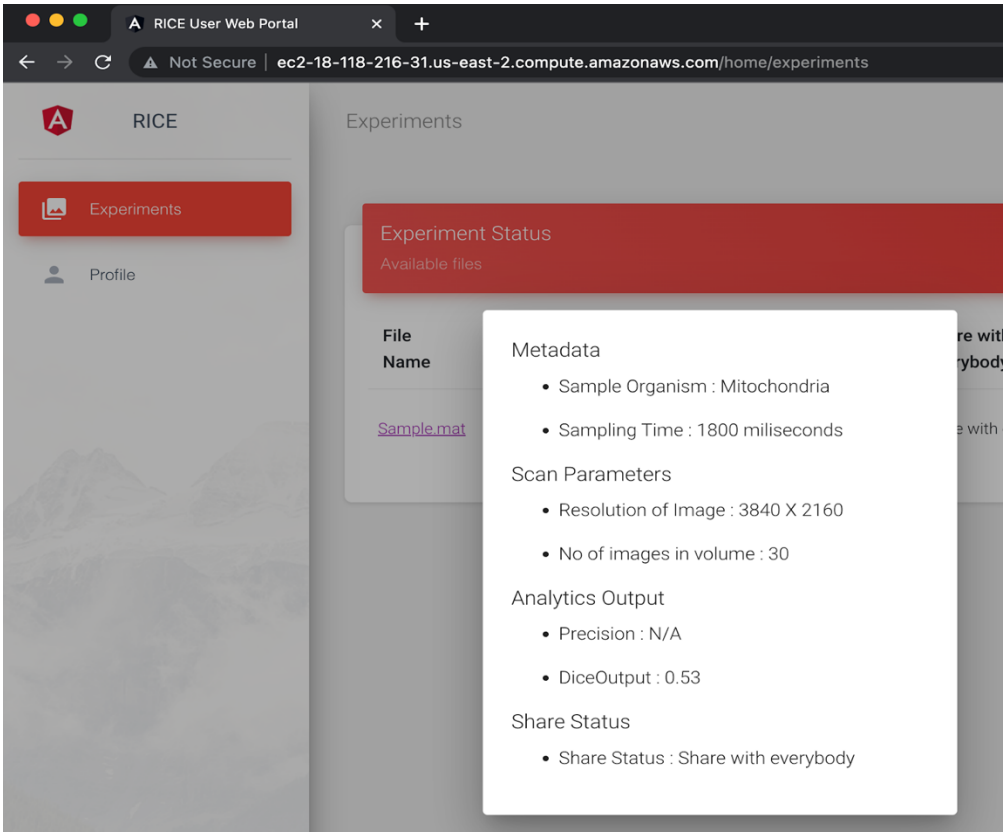

Figure 11 Captured image metadata

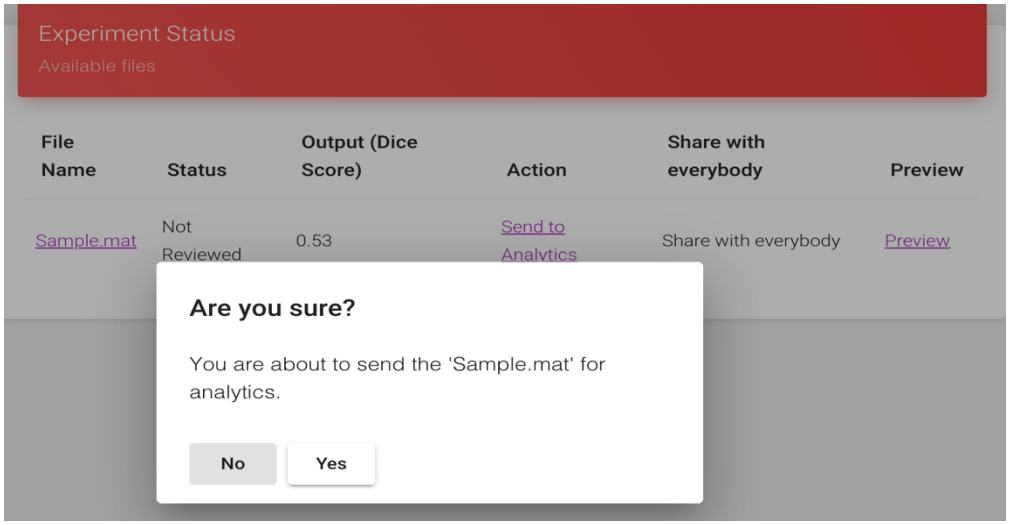

Figure 12 Confirmation Pop up

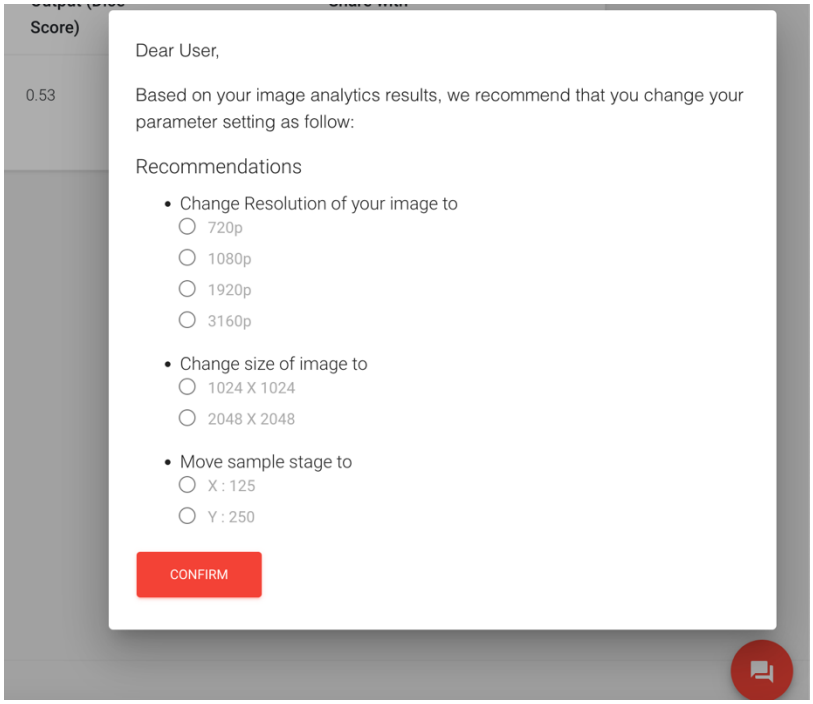

Figure 13 Recommendation Selection Screen after performing Image Analytics

8. Conclusion

This project allowed us to gain an insight regarding the high-level overview of how cloud related technologies can be implemented to solve real world problems. The goal of our project was to aid researchers in studying the mitochondria of the human cell and its implication on the future of healthcare. From a technological point of view, we were able to comprehend how remote instrumentation and AI can be integrated on sophisticated devices for automating the research process. Although the overall scope of the project was tremendous and required more time to complete, we were able to showcase the end-to-end interaction between different components and test the feasibility of image processing for automating the research. We believe that the current progress has laid out a framework for the continuation of this project in the future.

Working demonstration (Video Link):
https://www.kapwing.com/videos/627d728c7619e500c0845020

GitHub Repository: https://github.com/47carolines/Project2RICE