

Basic Details

- Suryansh Pathak
- Jaipur, India India Standard Time GMT +5:30
- 2nd Year BTech CSE Student at JKL
- Suryanshpathak5545@gmail.com & Suryansh5545
- live:suryanshpathak5545
- Resume:- [Drive](#)

Statement of Motivation

- What is your motivation for participating in Google Summer of Code?

I became interested to participate in GSOC after participating in GCI 2019 at EvalAI and getting to know what open-source organisation was. It was amazing at that time to make my first PR and get my code merged. GSOC will provide me with a structured environment and a clear objective and timeline. Which will also help me improve my time management skills. GCI provided me with experience and most important networking. I was able to meet a lot of exciting people still in touch with some I expect to make similar connections during GSOC. Also, it can't be denied that the stipend provided by GSOC will help me to focus my attention on the project solely.

- Why did you choose CloudCV?

I have previously worked on EvalAI during GCI and have a good understanding of how its Frontend works. It was a lot of fun working on the EvalAI frontend with CloudCV during GCI 2019. The mentors were very understanding and helpful and guided me throughout my GCI journey and beyond. At that time, one of the tasks I worked on involved making the UI between EvalAI and its ngx branch persistent, which introduced me to Angular and helped me develop my front-end development skills. Also, the mentorship and interaction I received during GCI helped me a lot in developing my communication skills. However, I was unable to continue working on EvalAI due to my 12th exam. Now that I have completed my 12th exam and got my college work together. I would like to return to working on EvalAI with CloudCV to further develop my skills and kickstart my career in AI. Working on EvalAI will require me to understand how various AI models work in sub-fields like Computer Vision, and NLP. For example, setting up a working demo of the VQA challenge for this idea will require me to understand how the model works.

- Why this project idea?

This project idea with gradio after implementation will allow user/participant to interact with their model more intuitively and with a user-friendly UI. It will also allow for the logging of adversarial data gathered at the time of interaction. This data can help in identifying weaknesses in the model. Using this data, the model can be made more accurate and ready for real world scenarios. This idea will allow me to learn about adversarial data and gain experience of using and integrating gradio.

- What are your expectations from us during and after successful completion of the program?

During the program, I look forward to gaining a deeper understanding of the evaluation process in EvalAI. I have found this to be a challenging area in the past. But I am ready to tackle it. I am hoping to have regular check-ins with my mentor to discuss any questions or concerns that I may have throughout the program. After the completion of the program, I hope to continue working on EvalAi and stay involved in the project.

- What are you hoping to learn?

By completing this proposed idea, I hope to learn about VQA. It is a interesting field with requirement of NLP and CV. Stuff like how it work. Also, I am looking forward to understanding Gradio platform by deploying it and integrating it into the EvalAi. This process will also including getting to know about its features and capabilities. I also hope to have a extensive understanding of the AWS RDS, SQS and EKS services. This project will be the first one in which I will get a hands-on experience of these services.

Experiences

- Have you taken Computer Vision, Machine Learning, Artificial Intelligence, Natural Language Processing, Deep Learning courses?

I haven't taken any courses in the above-mentioned fields. But I am in the middle of completing one at the time of writting this proposal. "Machine Learning A-Z: Hands-On Python & R In Data Science".

- What kind of projects have you worked in the past? What technologies did you use?

Electricity Bill Generation System with Invoicing(in House Project):- Java,Maven

Electrical Feeder Security System(Application Based) :- C++ and Javascript(For Application)

EvalAI :- AngularJs, Angular 6, Django, Python, AWS

- What is your experience with Python and Javascript?

I have a good understanding of Both. Python is used every day in my education field for 2 years now to do practicals on the mathematical concept of PDE, ODE, and Fourier Series these practicals include using libraries like matplotlib, scripy, and numpy. We have to plot graphs for comparison between given equations of FTCS, etc.

In JavaScript, I have good understand to connect how pages are working and debug it is required.

CloudCV Contributions

If you have contributed to projects of CloudCV, please provide the link of the pull requests with one-line description of the feature you added.

[EvalAI#3892](#) :- Added a Categories field with search capability (In Progress).

[EvalAI#3881](#) :- Adds a Cookie Consent to Frontend_v1 (Not Merged)

[EvalAI#3877](#) :- Added Search Bar to both Frontend_v2 and v1 (Reviewed, Not Merged)

[EvalAI-ngx#242](#) :- Add Url Submission for large file for participants (Merged)

[EvalAI-ngx#234](#) :-Adding google scholar field other evalai (Merged)

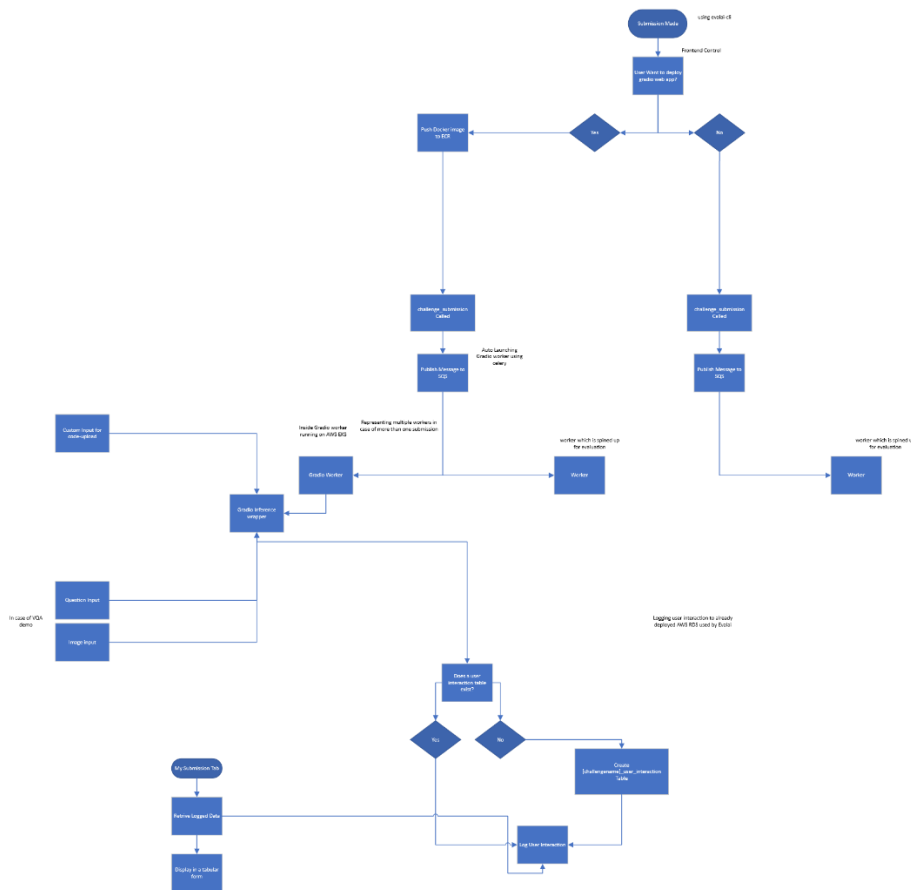
Other open-source contributions

I have contributed to the Submittly open source organisation, they have a Course Management software which functions similar to Canvas. The software is build on MVC software architecture and uses Php, Twig, Javascript, Typescript, Docker.

Project Details

The main objective of this project is to deploy user submissions on a Gradio service, enabling them to interact with it while logging the interaction details. These details can be stored in the same RDS used by the current Evalai infrastructure. The project will involve building a custom Gradio interface wrapper capable of handling modular input. Modular input implies that if a VQA challenge is being conducted, users should be able to provide image or text-based questions as input. The system's operation is depicted in the attached flowchart.

.Link to [Flowchart](#)



The already existing REST API which sends a message to the SQS Queue about the submission being made and will be called to deploy the Gradio worker. Gradio app will be in a dockerized container environment . The User interaction will be send by REST API and the interaction will also be send to the AWS RDS to a table possible called [challengeName]_user_interaciton.

To first collect the user interaction, we can use Fluentd, it can be configured to read actions written in a specific file. Since gradio doesn't have any logging of actions by default(There are some callbacks which do store the input and output but they can only be initiated by using the "Flag button") using the logging module of python. We can create a log file. Which can later be read by Fluentd and pushed to AWS RDS.

Example code to log user interaction.

```
import logging
import gradio as gr
# Set up logger
logger = logging.getLogger(__name__)
logger.setLevel(logging.INFO)
# Create file handler for the logger
fh = logging.FileHandler('interaction.log')
fh.setLevel(logging.INFO)
def vqa_interface(question: gr.inputs.Text, image: gr.inputs.Image):
    # Make prediction using model
    prediction = model.predict(question, image)
    # Log model prediction
    # Return model prediction as output
    log_dict = {
        'question': question,
        'image': image,
        'prediction': prediction,
        'timestamp': time.time()
    }
    log_message = json.dumps(log_dict)
    logger.info(log_message)
    return prediction
# Run Gradio app
gr.Interface(fn=vqa_interface, inputs=["question", "image"], outputs="text").launch()
```

Later, when the user wants to download the logged interactions, they can be retrieved from AWS RDS by the Django backend and stored in a CSV file.

The option to download the CSV file can be shown in the My Submission tab.

Like this example below

OverviewEvaluationPhasesSubmitMy SubmissionsLeaderboard

My Submissions

My Participated Team: Rosstown Participant Team

Jacob Phase

File type

Download

Total Submissions: 10

ation URL	Status	Execution time (sec.)	Submitted file	Result file	Stdout file	Stderr file	Submitted at	Show on leaderboard	Edit	Cancel	Gradio Interaction Log
ww.hayes-smith.biz	Finished	0.000002	Link	None	None	None	Mar 31, 2023 12:08:46 PM	✓			Link
asley.info/	Finished	0.000002	Link	None	None	None	Mar 31, 2023 12:08:46 PM	✓			Link
olloway-hancock.comry.htm	Finished	0.000001	Link	None	None	None	Mar 31, 2023 12:08:46 PM	✓			Link

And when the user is submitting their model from the frontend, they can choose to have a Gradio web application deployed by showing them an option at the time of submission.

Dashboard

All Challenges

Hosted Challenges

Create Challenge

Participant Teams

Host Teams

Make Submission

Select phase:

☐ Phase: Crystal Phase
Start date: Dec 21, 2022 12:08:45 PM IST
End date: Aug 12, 2024 12:08:45 PM IST

☒ Phase: Jacob Phase
Start date: Dec 21, 2022 12:08:45 PM IST
End date: Aug 12, 2024 12:08:45 PM IST

Today's remaining submissions: 99995

Monthly remaining submissions: 99995

Total remaining submissions: 99995

Select submission type:

☐ Use CLI (for file size > 400MB)

☐ File URL (for file size > 400MB)

☒ Upload file

Upload file

Select submission visibility:

☒ Public

☐ Private

☐ Gradio Web App Deployment

Submit

Deliverables: -

1. Build system design for EvalAI and Gradio integration.
2. Set up Gradio integration for a single VQA demo as a proof of concept, including:
 - Creating a Gradio interface wrapper for model inputs.
 - Integrating the Gradio interface with a dockerized model for inference.
3. Set up auto-launching of a worker service for deploying a Gradio app using a celery task from EvalAI's Django backend.
4. Add frontend controls to deploy a Gradio web service for a single submission made by a participant to a code upload challenge.
5. Add support to log interactions made by a user on the Gradio web service and push the interactions to a database table specific to a demo.
6. Add API support to export the logged interactions for a demo.
7. Implement frontend changes to allow users to download the logged interactions in a standard dataset format.
8. Add test cases for the newly created Gradio worker.
9. Edit test cases for the changes made in Frontend_v1.

Extended Project Idea

I want to further extend this project with the task of Integrating it with Huggingface platform. This will allow user to submit his pretrained models from huggingface if he have it on the platform. We can also allow challenge host to download the dataset from Huggingface platform. Challenge host can also decide to use a Huggingface model as a baseline.

Project Schedule: -

Official Coding Period

- 28th May to 4th June Week 1: - Modifying the Frontend and building system design for Evalai and Gradio Integration.

- 4th June to 18th June Week 2-3: - PR to Create the Gradio Interface frontend, backend for modular input.
- 18th June to 25th June Week 4: - Integrating the Gradio interface to challenge_submission pipeline to deploy it on AWS EKS.
- 25th June to 9th July Week 5-6: - Implementation of data logging and Fluendt to push it to the pre-existing AWS RDS.
- Mid Term Evaluation.
- 9th July to 16th July Week 7: - Implementing frontend changes to the My Submission tab to download the user interactions.
- 16th July to 23rd July Week 8: - Adding Test cases for gradio interface.
- 23rd July to 30th July Week 9: - Adding Test case for the Frontend modifications.
- 30th July to 6th August Week 10: - Deploying VQA demo as a proof of concept.
- 6th August to 21st August Week 11-12: - Finishing up the rest of the work and any changes required after the proof-of-concept test.
- End Term Evaluation.

In the Community Bonding Period, I hope to familiarize myself in depth with evalai evaluation process and how the worker is deployed after the submission is made, I would also like to understand in depth how after submission the SQS queue deploys the worker and how the auto scaling works in all this.

After GSOC period ends, I would like to continue working on the Evalai, possibly expanding on my Huggingface idea beyond the submission feature. Furthermore, I would like to represent CloudCV as mentor in GSOC 2024.

Time: -

I will have my End Term exam from 9th to 12th May after which, I will be on break till 1 August. So, I will be able to dedicate more than 40 Hour a week.

Furthermore, I would like to start the work on my PR before 28th May. So, I can have extra time for all the unforeseen issues that might arise during the implementation of these features. Furthermore after the opening of my College on 1st August I will still be able to give time to the project which will be more than enough to achieve the objective and possibly go beyond it.