Proposal For GSoC 2023

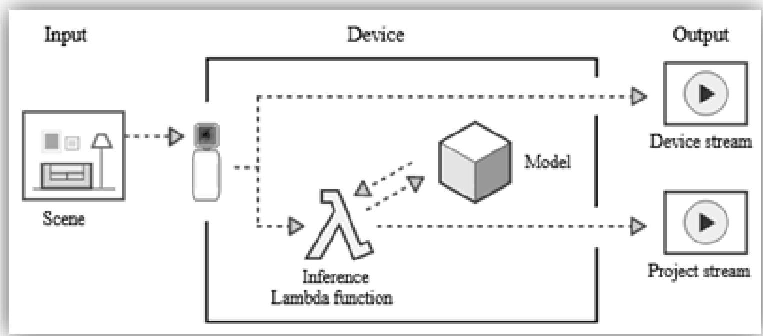
CONTACT DETAILS

Name:	Shibam Nath
Nickname:	Subho
Country:	India
Timezone:	IST (India)
Typical Working Hours:	11:30-13:30, 15:00-19:00, 22:00-05:00 (IST) 6:00-8:00, 9:30-13:30, 16:30-23:30 (UTC)
Email Id:	shibamnath12032002@gmail.com
Phone:	+91 6289725774
GitHub:	https://github.com/shibam120302
LinkedIn:	https://www.linkedin.com/in/shibam-nath-0a23a6227/
Language:	English
Linktree:	https://linkpop.com/shibamnath

Project

Project Name

DeepLense is a deep learning pipeline for particle dark matter searches with strong gravitational lensing.



Project Description https://github.com/shibam120302/GSoC23_ML4SC_Lens_Finding

DeepLense is a project that aims to develop a deep learning pipeline for particle dark matter searches using strong gravitational lensing. The project involves using machine learning techniques to analyze astronomical images and identify strong gravitational lenses, which are regions of space where the gravitational pull of massive objects (such as galaxies) bends and distorts light from more distant objects.

The project has several key objectives. First, the team will work on developing a deep learning model that can accurately identify strong gravitational lenses in astronomical images. This model will be trained on a large dataset of simulated images, using a variety of different architectures and training techniques to achieve the best possible performance.

Next, the team will work on integrating this model into a larger pipeline for particle dark matter searches. This pipeline will involve using the identified gravitational lenses to infer properties of dark matter particles, such as their mass and distribution. The pipeline will be optimized for performance and accuracy, and will be designed to be user-friendly and easy to use for researchers in the field.

Finally, the team will work on developing tools and documentation to make it easy for other researchers to use the DeepLense pipeline in their own work. This will involve creating user-friendly interfaces for configuring and running

the pipeline, as well as providing detailed documentation and tutorials on how to use the pipeline effectively.

To achieve these objectives, the team will use a combination of deep learning libraries (such as TensorFlow or PyTorch), astronomical image processing libraries (such as Astropy), and software development tools (such as Git and Jupyter notebooks). They will also take advantage of the resources and support provided by the GSoC program, including mentorship from experienced researchers and access to computing resources.

Overall, the DeepLense project has the potential to make a significant contribution to the field of particle dark matter searches. By developing a deep learning pipeline that can analyze strong gravitational lenses and infer properties of dark matter particles, the project could help researchers better understand the nature of dark matter and its role in the structure and evolution of the universe.

Predesign

Notebook Link : rnn_ring_classification

Objective:

Develop a deep learning pipeline for particle dark matter searches using strong gravitational lensing in astronomical images.

Key tasks:

1. Collect a large dataset of simulated astronomical images containing strong gravitational lenses and non-lens objects.

Download and extract data

The training sets from the Strong Lensing Challenge can be downloaded from the challenge page with the following procedure:

```
$ cd [data_dir]
$ wget http://metcalf1.difa.unibo.it/blf-portal/data/GroundBasedTraining.tar.gz
$ tar -xvzf GroundBasedTraining.tar.gz
$ cd GroundBasedTraining
$ tar -xvzf Data.0.tar.gz
$ tar -xvzf Data.1.tar.gz
```

2. Once the is downloaded we use the following script to turn it into a convenient astropy table.

```
from astropy.table import Table
import pyfits as fits
import numpy as np

# Path to the downloaded files
download_path=[data-dir]  # To be adjusted on your machine

# Path to export the data
export_path=[data-dir]   # To be adjusted on your machine

# Loads the catalog
cat = Table.read(download_path+'GroundBasedTraining/classifications.csv')

ims = np.zeros((20000, 4, 101, 101))

# Loads the images
for i, id in enumerate(cat['ID']):
    print i
    for j, b in enumerate(['R', 'I', 'G', 'U']):
        ims[i, j] = fits.getdata(download_path+'GroundBasedTraining/Public/Band'+str(j+1)+'/imageSDSS_'+b+'-'+str(id)+'.fits')

# Concatenate images to catalog
cat['image'] = ims

# Export catalog as HDF5
cat.write(export_path+'catalogs.hdf5', path='/ground', append=True)

print "Done !"
```

3. Develop a deep learning model that can accurately identify strong gravitational lenses in astronomical images, using convolutional neural networks (CNNs) and transfer learning techniques.

```python
from astropy.table import Table

# Loads the table created in the previous section
d = Table.read('catalogs.hdf5', path='/ground')     # Path to be adjusted on your machine
```

```python
# We use the full set for training,
# as we can test on the independent challenge testing set
x = array(d['image']).reshape((-1,4,101,101))
y = array(d['is_lens']).reshape((-1,1))
# [Warning: We reuse the training set as our validation set,
# don't do that if you don't have an independent testing set]
xval = array(d['image'][15000:]).reshape((-1,4,101,101))
yval = array(d['is_lens'][15000:]).reshape((-1,1))
```

```python
# Clipping and scaling parameters applied to the data as preprocessing
vmin=-1e-9
vmax=1e-9
scale=100

mask = where(x == 100)
mask_val = where(xval == 100)

x[mask] = 0
xval[mask_val] = 0

# Simple clipping and rescaling the images
x = np.clip(x, vmin, vmax)/vmax * scale
xval = np.clip(xval, vmin, vmax)/vmax * scale

x[mask] = 0
xval[mask_val] = 0
```

```python
# Illustration of a lens in the 4 bands provided
im = x[0].T
subplot(221)
imshow(im[:,:,0]); colorbar()
subplot(222)
imshow(im[:,:,1]); colorbar()
subplot(223)
imshow(im[:,:,2]); colorbar()
subplot(224)
imshow(im[:,:,3]); colorbar()
```

4. Train and optimize the deep learning model on the dataset of simulated images, using techniques such as data augmentation, regularization, and hyperparameter tuning.
5. Evaluate the performance of the deep learning model on a validation set of simulated images, and iterate on the model architecture and training process as necessary to improve performance.

6. Develop a pipeline that can process astronomical images and identify

strong gravitational lenses using the trained deep learning model, including pre-processing steps such as image segmentation and feature extraction.

7. Integrate the pipeline with tools for particle dark matter searches, such as tools for estimating the mass and distribution of dark matter particles based on the identified gravitational lenses.

```
def plot_accuracies(history):
  accuracies = [x['val_acc'] for x in history]
  plt.plot(accuracies, '-x')
  plt.xlabel('epoch')
  plt.ylabel('accuracy')
  plt.title('Acc vs Epochs')

def plot_loss(history):
  loss = [x['val_loss'] for x in history]
  plt.plot(loss, '-x')
  plt.xlabel('epoch')
  plt.ylabel('loss')
  plt.title('Loss vs Epochs')

def plot_lr(history):
  lr = [x['lrs'] for x in history]
  plt.plot(lr, '-x')
  plt.xlabel('epoch')
  plt.ylabel('learning_rates')
  plt.title('Learning_rates vs Epochs')
```

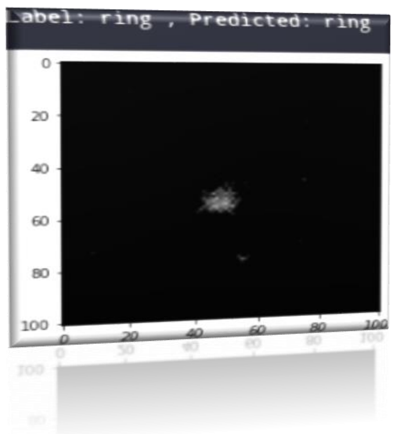
8. Develop user-friendly tools and documentation to make it easy for other researchers to use the DeepLense pipeline in their own work, including tutorials, sample scripts, and user interfaces for configuring and running the pipeline.

Technologies and tools:

- Deep learning libraries such as TensorFlow or PyTorch for developing and training the deep learning model.
- Astronomical image processing libraries such as Astropy for pre-processing and analysis of astronomical images.
- Software development tools such as Git for version control and collaboration.
- Cloud-based computing resources such as Google Cloud or Amazon Web Services for training the deep learning model on a large dataset.

Potential challenges:

- Limited availability of high-quality simulated images of strong gravitational lenses, which may require developing custom simulation tools or using alternative approaches such as data augmentation to expand the dataset.



- Difficulty in identifying and separating strong gravitational lenses from other objects in astronomical images, which may require developing more sophisticated image processing and feature extraction techniques.
- High computational requirements for training deep learning models on large datasets, which may require using cloud-based resources or optimizing the model architecture and training process to reduce training time and resource usage.

Timeline

COMMUNITY BONDING PERIOD (MAY 4 – MAY 28)
• Introduce myself and this project in Develop a deep learning pipeline mailing list.
• Remain constant touch with my mentors using Google Hangouts. Set up user requirements and discuss the design details with mentors. Settle the final design according to user requirements and feasibility.
• Discuss with mentors about the implementation plan.
• Try to fix bugs to get further understanding for deep learning source code.
• Set up dev environment and my wiki page for TODO list and weekly report.

OFFICIAL CODING PERIOD (MAY 29 - AUGUST 28)

May 29 – June12 (2 weeks)

- Familiarize with the project and set up development environment
- Research existing datasets and simulation tools for creating simulated astronomical images
- Develop a plan for creating a large dataset of simulated astronomical images containing strong gravitational lenses and non-lens objects

June 12- June 26 (2 weeks)

- Develop a deep learning model architecture that can accurately identify strong gravitational lenses in astronomical images, using convolutional neural networks (CNNs) and transfer learning techniques
- Train the deep learning model on a subset of the simulated dataset
- Develop a validation set of simulated images for evaluating the model's performance

June 26 - July 10 (2 weeks)

- Optimize the deep learning model's performance by iteratively improving the model architecture and training process, using techniques such as data augmentation, regularization, and hyperparameter tuning
- Evaluate the model's performance on the validation set and refine the model accordingly

July 10 - July 24 (2 weeks)

- Develop a pipeline that can process astronomical images and identify strong gravitational lenses using the trained deep learning model, including pre-processing steps such as image segmentation and feature

extraction
- Integrate the pipeline with tools for particle dark matter searches, such as tools for estimating the mass and distribution of dark matter particles based on the identified gravitational lenses

July 24 – Aug 7 (2 weeks)

- Develop user-friendly tools and documentation to make it easy for other researchers to use the DeepLense pipeline in their own work, including tutorials, sample scripts, and user interfaces for configuring and running the pipeline
- Write detailed documentation and prepare a user guide for the pipeline

August 7 - August 21 (2 weeks)

- Test the pipeline on additional simulated datasets and real-world astronomical images
- Refine the pipeline based on feedback from users and performance testing

August 15 - August 22 (1 weeks)

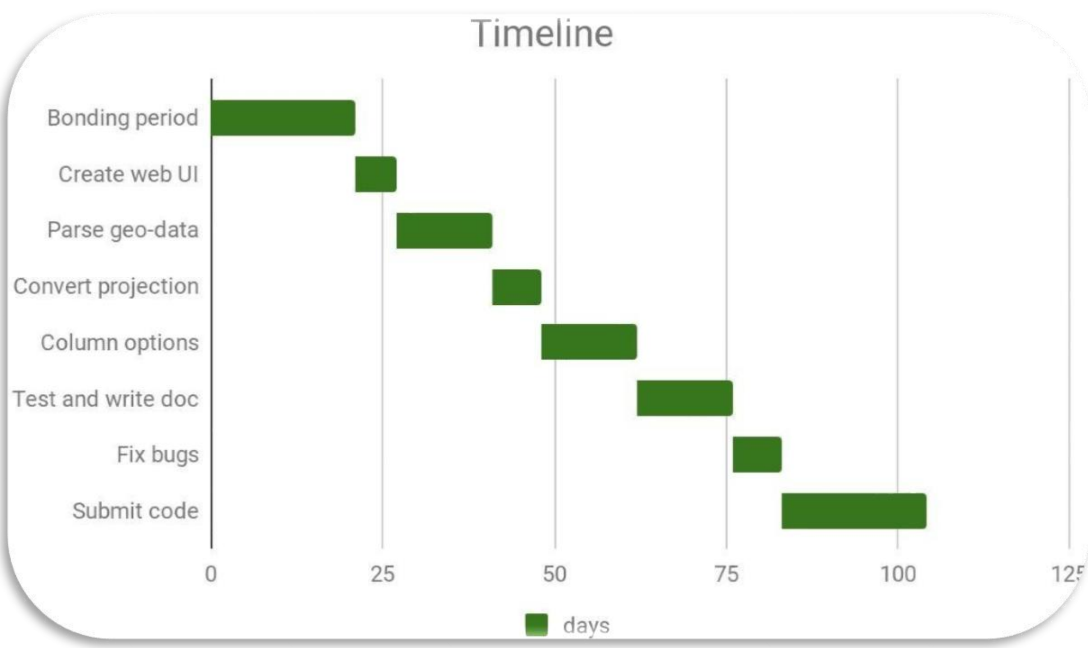
- Prepare the final report summarizing the project and its results
- Finalize the documentation and user guide for the pipeline
- Present the project in a final presentation to the community

August 22 - September 28 (6 days)

- Buffer time for any pending work or improvements to the project
- Submit the final project report and code repository

Deliverables

A large dataset of simulated astronomical images containing strong gravitational lenses and non-lens objects
A deep learning model that can accurately identify strong gravitational lenses in astronomical images, trained on the simulated dataset
A pipeline that can process astronomical images and identify strong gravitational lenses using the trained deep learning model, integrated with tools for particle dark matter searches
User-friendly tools and documentation for using the DeepLense pipeline in other research projects
A final project report summarizing the project's objectives, methods, results, and future work



Participation

Progress Report

- I will remain online on IRC, hangouts in my working hours (1pm to 11pm UTC +5:30)
- I will write weekly blog posts at (https://shibamnath.blogspot.com/)
- I will share my blogs on twitter
- Write weekly scrum reports and update it to our mailing list
    - What did I do last week?
    - What will I do this week?
    - What is currently preventing me from reaching goals?

- I will submit a Project Presentation

Where I plan to publish my source code

- I will be working on a separate branch on git and uploading code to the forked repository almost on a daily basis, will be Creating pull requests when a complete feature is done.

Communication on task

- I will use GitHub to manage bugs and task.

Do you understand this is a serious commitment, equivalent to a full-time paid summer internship or summer job?

> Yes, I understand. I will commit my full time in this project and can spend at least 40 hours per week. I have great passion in coding and implementing the data viewer. I am willing to pay my efforts in making the tool more convenient to use.

Do you have any known time conflicts during the official coding period?

> No, I don’t have any time conflicts with the official coding time

Extra Information

Studies

I am a second-year B. Tech. Undergraduate at Indian Institute of Engineering Science and Technology, Shibpur, pursuing Electronics and Telecommunication Engineering as my Major and interested in machine learning and deep learning.

This project will definitely contribute to my study because it will not only make it more convenient for me, but also help me improve skills in, coding, writing documentation and getting involved in open source community.

Programming and Deep Learning

I am familiar with Python, C/C++, Java and have the basic knowledge of html, JavaScript and CSS. Also, I have the experience in using Flask framework and OpenLayers. I use Window 10 and Ubuntu 16.04 LTS for daily basis.

I have completed several courses in machine learning and deep learning, including courses on Coursera and Udemy. I have experience with various machine learning algorithms such as decision trees, support vector machines, neural networks, and deep learning frameworks such as TensorFlow and Keras.

In addition, I have worked on several machine learning projects, including developing a fraud detection system for a financial institution and a natural language processing model for sentiment analysis. I have also worked on research projects in the field of deep learning, including developing a convolutional neural network for image classification. Here are ***my two personal deep learning projects:***

1. Pancard Tempering Detector App

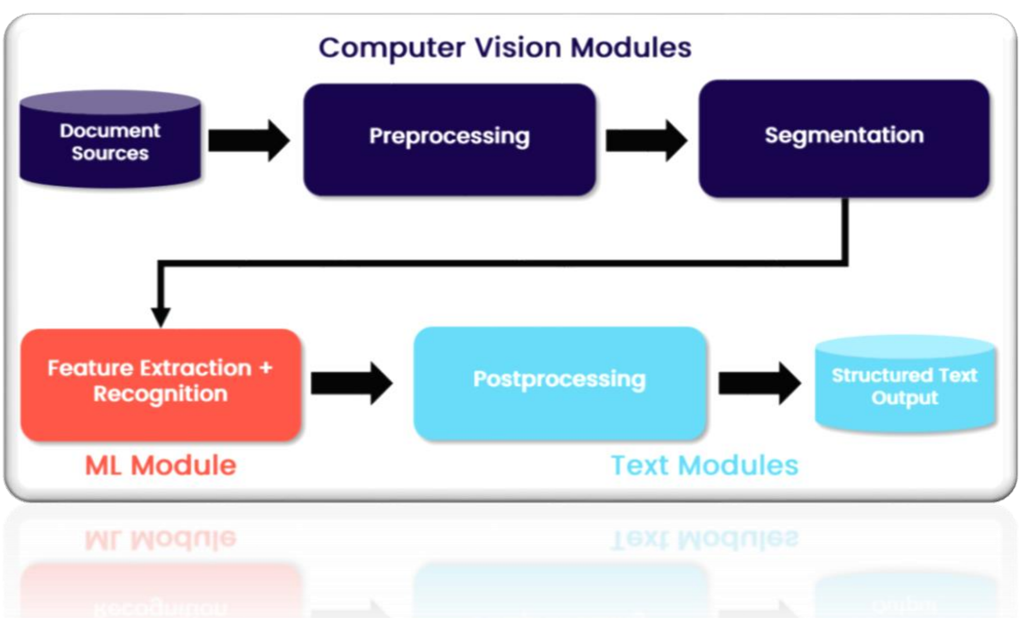
Purpose

The purpose of this project is to detect tampering/fraud of PAN cards using computer vision. This project will help the different organizations in detecting whether the Id i.e. the PAN card provided to them by their employees or customers or anyone is original or not.

For this project we will calculate the structural similarity of the original PAN card and the PAN card uploaded by the user – This is the soul of this project we will discuss it thoroughly later in this blog.

Similarly in this project with the help of image processing involving the techniques of computer vision we are going to detect that whether the given image of the PAN card is original or tampered (fake) PAN card.

2. Text Extraction Flask App



Purpose

The purpose of a Text Extraction Flask App is to extract relevant information or data from a given text or document. This can be particularly useful in situations where a large amount of unstructured text needs to be analyzed or processed, such as in natural language processing, data mining, or information retrieval.

GENERAL QUESTION – ANSWER

How did I hear about this program ?

I heard about GSoC an year back in a college meetup. I am an open source enthusiast from past 1 year and I always wanted to take part in Google Summer of Code.

Time during Summers

I have no other commitments this summer. So I'll be able to give 40 hours or more per week. I am ready to commit extra time if needed in order to finish up the goals of the project. My summer break starts from May, so I can start working full time from that day on. I'll not be taking any vacations. My classes start on around 20th July but I will be able to commit enough time for the project as there are no exams during the period.

Eligible for Google Summer of Code and Outreachy ?

I am applying only for GSoC as I am not eligible for Outreachy program. I am applying under Wikimedia Commons Android only.

What excites me about this project?

The kind of exposure and experience a platform like this would provide me is a huge reason for me to want to be a part of it. I have developed a lot of apps during my college days, but I always wanted to develop apps that really help people at a global scale. I find nothing more exciting than working for a company like Google with a goal as impactful as this one. I sincerely love the goal of Wikimedia Foundation "Global movement whose mission is to bring free educational content to the world" and would be more than happy to work towards it. It would be really great for me to apply my skills and contribute to such an organization.

Why should I be selected for the project ?

I have always been interested in open-source projects and have been passionately working on them. I can push myself to the boundaries, come out of my comfort zone and work things out. I am a very good team-player and can learn thing quickly and adapt myself. I have been developing Flask apps in ML for more than 2 years now and having contributed to various open source projects, I believe I have the required skills to finish the proposed goals of the proposal. Moreover, having enrolled in Google's Machine learning program, I believe I will follow the best practices in implementing the goals of the proposal. I have been contributing to ML4SCI GitHub for over 45 days now and will be contributing to it even after the GSoC period.

GSoC participation

This is my first time to apply for Google Summer of Code. I did not submit a proposal to any other organisation.