Google Summer of Code

Machine Learning for Science

GSoC 2023
“Transformers for Dark Matter Morphology with Strong Gravitational Lensing”

Project Proposal
By
Neeraj Anand

Table of Contents

About Me

Student Information

Name:	Neeraj Anand
Email:	neerajanandfirst@gmail.com, 20je0609@am.iitism.ac.in
TimeZone:	New Delhi, India (+5:30 GMT)
Github:	neerajanand321
LinkedIn:	neeraj-anand-092353204/
Resume:	Link
Kaggle:	Kaggle _profile
Address:	Supaul, Bihar, India Pin- 852108

University Information

University:	Indian Institute of Technology (ISM) Dhanbad
Major:	Mathematics and Computing
Current year:	3rd
Expected Graduation date:	August 2025
Degree:	Integrated M.Tech

Background work and Programming Skills

- I am a third-year undergraduate student of IIT (ISM) Dhanbad, India. I'm pursuing a degree in Mathematics and Computing.
- I use a Macbook for work, Google Colaboratory, Jupyter and Kaggle notebook for implementation and VSCode for development and vim for SSH sessions.
- I am proficient in Python, C++ and PyTorch.
- I am a passionate machine learning researcher with valuable skills in research methodology, data analysis, and scientific writing from conducting original research during a fulfilling research internship. I have also worked on a research paper MixBin: Towards Budgeted Binarization, The work is almost complete and we are targeting NeurIPS'23.
- I am also an active participant in Kaggle competitions, where I have been able to apply my machine learning knowledge to solve real world problems. I have been awarded the bronze medal in the G2Net detecting continuous gravitational waves. This competition required participants to employ deep learning models to accurately detect long-standing waves. My solution can be found here
- I know how to use Git and Github.
- If I am stuck, I go to Google and always come back with a solution.

Evaluation Task

I have successfully completed all the mandatory tasks listed on the website, as per the given guidelines. I am particularly delighted to report that I have achieved a remarkable roc_auc score of 0.98 for the Common Task, and an equally impressive score of 0.98 and 0.99 for Specific Task 2 and Specific Task 5, respectively, also trained a model to learn the distribution of the provided strong lensing images with no substructure. Furthermore, I would like to share that the GitHub repository containing the solution to the aforementioned tasks can be accessed through the link

Project Information

Organization: ML4SCI

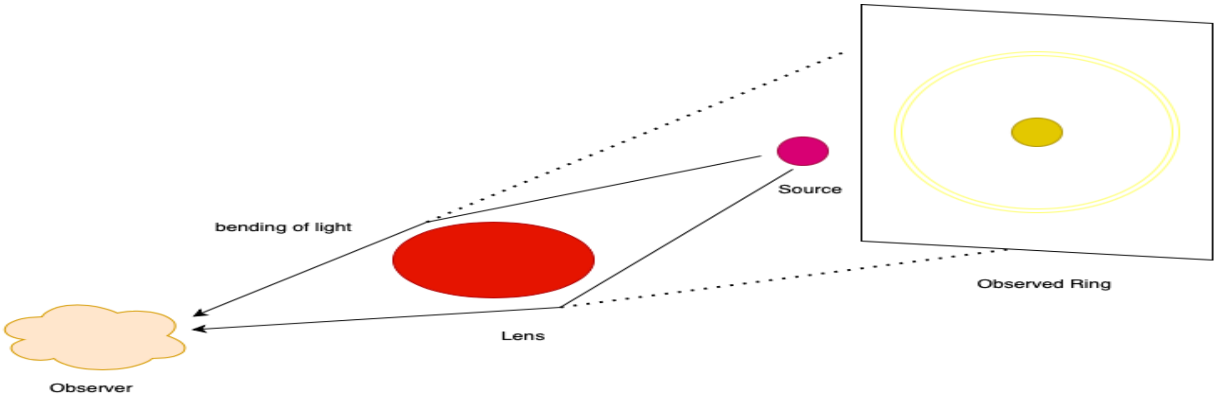
(ML4SCI_Org)

Project : Gravitational Lens Finding for Dark Matter Substructure Pipeline

Possible mentor: Emanuele Usai, Anna Parul, Micheal Toomey, Pranath Reddy, Stephon Alexander, Saranga Mahanta, Karthik Sachdev

Gravitational lensing is a phenomenon in which the path of light from a distant object, such as a galaxy or a quasar, is bent by the gravitational field of a massive object, such as a galaxy cluster or a black hole.

The gravitational field of the massive object (Gravitational lens) acts as a lens that can magnify, distort, or even multiply the image of the distant object.



The lensing are of two types: Strong Lensing and Weak Lensing, For strong lensing, the projected lens mass density must be greater than critical density. This strong lensing can help us to find the substructure of dark matter. As dark matter has mass but doesn't emit light or electromagnetic waves, it is difficult to study directly. However, it can bend light and act as a lens, which makes it identifiable by gravitational lensing. Deep learning methods have the potential to accurately identify images containing strong Lenses.

Abstract

The DeepLense Pipeline leverages the use of deep learning to identify strong lenses from real and simulated images based on Lenstronomy in a dataset .
This project aims to expand the DeepLense functionality to include vision transformers suitable for computer vision tasks applicable to DeepLense data.

For this project, I would analyze the most recent DeepLense data sets, develop a deep vision transformer learning model for DeepLense training and inference.

Objective / Idea

I will be working this summer on,

- Analysis of Dataset.
- Exploring various augmentation to be used on dataset. (Try some Custom augmentation also).
- Exploring a wide range of deep vision transformer architectures.
- Training Methodology eg. Training from scratch vs using Pretrained weights, Distillation signal.
- Choice of optimizer, scheduler, loss functions etc.
- Tuning of hyperparameters.
- Develop deep vision transformer models for other tasks eg. anomaly detection.
- Ensemble of several Deep Learning Models.
- Documentation of the codebase for helping other open-source enthusiasts contribute easily.
- If time permits, work on compression of trained networks by applying binarization, pruning and quantization.

In my view, the relevant skills required would be Python, PyTorch, and knowledge of Machine Learning and Deep Learning.

Implementation/Approach

WorkFlow :

1. Analysis of Dataset :

For any Machine Learning task, one of the most important things is to analyze the data and gain as much information as we can. I will study about the class imbalance, similarity between the real and simulated data and simulation of more data if required.

How will it help ?
It will help to build a better and more robust model.

How will I implement it ?
We'll create a separate notebook for the analysis part. For finding the similarity between the real and simulated data first will study the attributes of the real data and the hyperparameters used to create the simulated ones and then try to tune the value of hyperparameters to reduce the gap.

2. Train a Baseline ML Model

To ensure high accuracy in object classification and detection tasks, it is important to train a baseline model on a dataset that is representative of the target domain. Therefore, I intend to train a baseline model on the most recent DeepLense dataset.

How will it help?
Help to establish a benchmark performance and evaluate the effectiveness of future improvements.

How will I implement it?
Will create transformer.py in models folder, define the baseline deep vision transformer models and the pretrained timm models then start by training the baseline network. Once the baseline model has been trained, I will move on to more advanced models such as pretrained vision transformer models like VIT and SwinTransfromer and their different versions.

3. Study of Hyperparameters, Models to reduce the overfitting, underfitting and to stabilize the training.

Proper tuning of hyperparameters is crucial to achieving good model performance. Common hyperparameters include learning rate, batch size, number of epochs, and regularization strength. Will use grid search or random search to tune these hyperparameters.

To reduce overfitting, I will use data augmentation, early stopping during the training process.

To stabilize the training process, I will find the appropriate value of learning rate, and will use gradient clipping.

How will it help?
Will help to design a more robust model.

How will I implement it?
We'll use Albumentation library to experiment with data augmentation,
In training folder, will create three files namely config.py, utils.py and training.py where config.py is used to change hyperparameters, optimizer, scheduler, loss function utils.py contains other helper functions and training.py contains class responsible to train and evaluate model

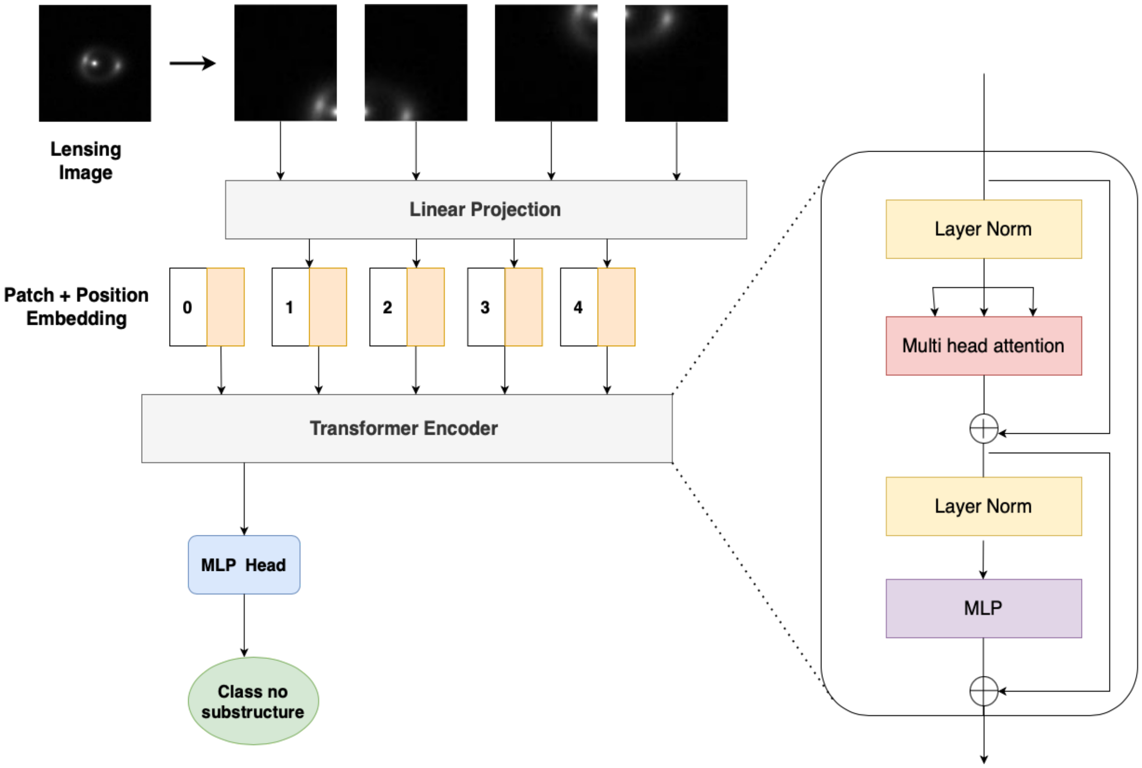
4. Domain Specific Research

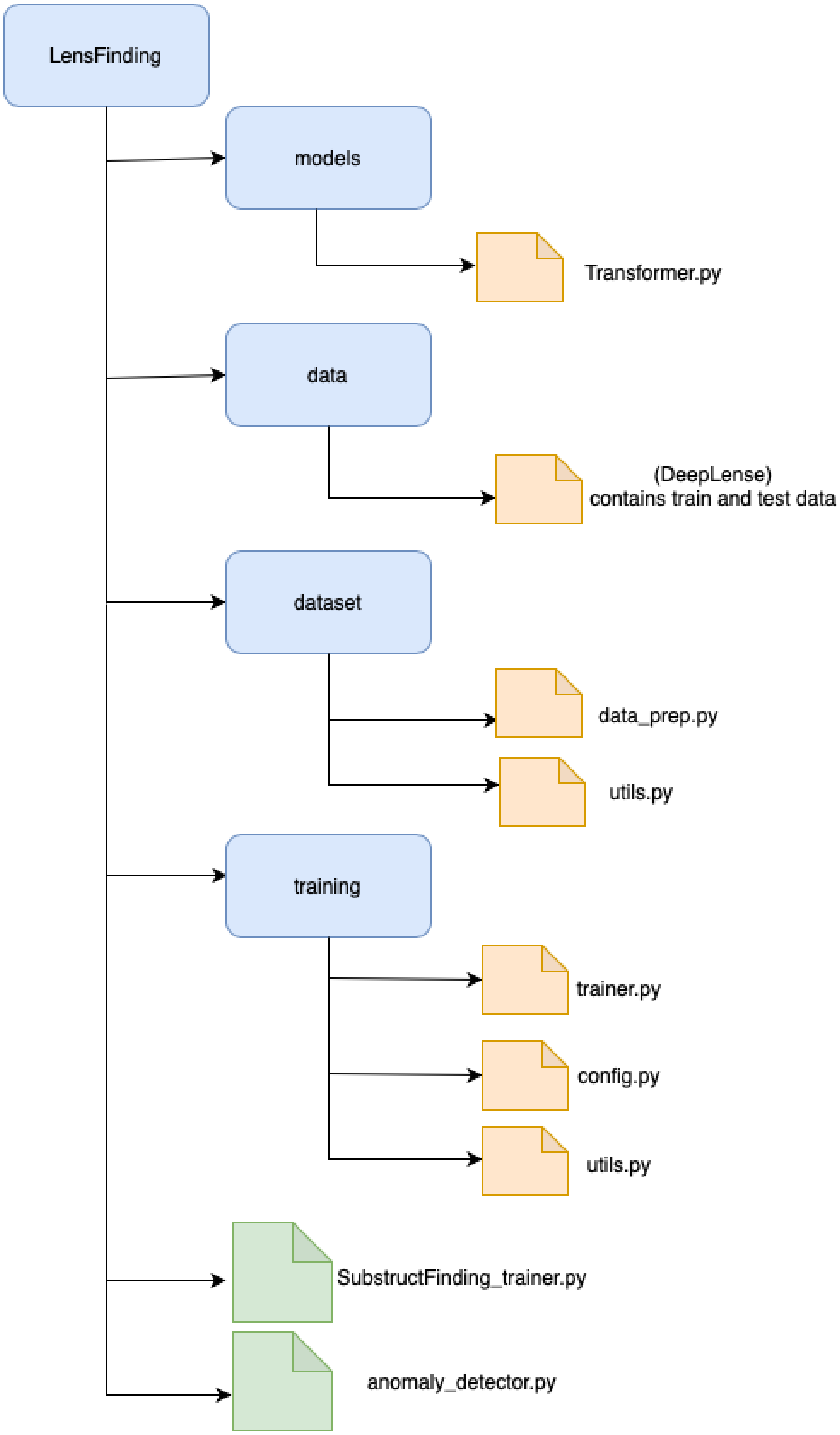
I intend to undertake a comprehensive evaluation of the model's performance across all types of gravitational lensing phenomena. Given that gravitational lenses can manifest as rings, arcs, or multiple images, the model will be penalized for its ability to accurately detect the presence of substructure. Of particular interest is **the performance of the model in the case of strong lensing due to point-like background source**.

How will it help?
Help to detect substructure for those gravitational lenses when there is a point-like background source.

How will I implement it?

To enhance the model's ability to accurately detect images with substructure, we will leverage data augmentation techniques. **I will train a specialized model that can detect the present substructure.** To further improve the model's performance, we will implement a **progressive learning approach** and remove easy negative samples to encourage the model to learn more complex and challenging samples.




LensFinding
models
Transformer.py
data
(DeepLense)
contains train and test data
dataset
data_prep.py
utils.py
training
trainer.py
config.py
utils.py
SubstructFinding_trainer.py
anomaly_detector.py

Timeline (Tentative)

From my past experiences, I can say that things can change in a minimal time. For example, some bug comes up with a high priority level which needs to be fixed before any more work. However, having a timeline is always suitable for any project. This, at least, helps in keeping a tab on how much you are lagging.

Pre-GSoC period

Before the coding begins in May last week, there is a lot of homework for me to do. I have to study gravitational lensing and how I can utilize my deep learning skills to detect it.

Community Bonding

During this period, I shall study the related research papers and start working on the codebase. I shall discuss new approaches regarding different algorithm implementations. I will be keeping notes too! I shall also discuss the further plans regarding the scalability and other relevant topics regarding the same.
The plan for GSoC 2023 will be :
I am planning to split the project into three phases :

- Phase 1: Analysis of dataset and building a baseline deep vision transformer model for different tasks.
- Phase 2: Improvement of the model on all the tasks, domain specific research.
- Phase 3: Project Setup Cleanup and Documentation: Help new contributors grasp quickly and contribute.

Timeline

Period	Task
Week 1 [May 29 - June 3]	• Load the dataset. • Setup the environment on the available resource. • Start developing the codebase.
Week 2 [June 4 - June 10] Week 3 [June 11 - June 17]	• Analysis of available dataset. 1. Study about skewness of dataset. 2. Similarity between real and simulated data. 3. Study of meta data if available. 4. Write data_prep.py and utils.py in the dataset.

Week 4 [June 18 - June 24] Week 5 [June 25 - July 1]	● Train a baseline model. 1. Write all files in the model folder (self defined as well as pre trained deep vision transformers). 2. Write all files in the training folder. 3. Write SubstructFinding_trainer.py 4. Write anomaly-detector.py 5. Start training the model on different tasks such as classification, anomaly detection. 6. Evaluation of the model on various metrics. 7. Add Documentation
Week 6 [July 2 - July 8]	● Study the effects of hyperparameters : 1. Try different network architecture. (VIT, Swin and their different versions) 2. Choice of optimizer. 3. Choice of scheduler. 4. Choice of Loss function.
Mid Evaluation [July 10 to July 14]	
Week 7 [July 16 - July 22] Week 8 [July 23 - July 29] Week 9 [July 30 - August 5]	● Study the effects of hyperparameters. 1. Tune learning rate, weight decay and other hyperparameters using grid search (Optuna). 2. Effect of gradient clipping, early stopping. ● Try different data augmentations. 1. Use the Albumenataion library for data augmentation. 2. Try Custom augmentation (self-defined). ● Add Documentations.
Week 10 [August 6 - August 12]	● Domain specific research. 1. Study the performance of the model on data due to point-like background sources. 2. Train the model using progressive learning techniques.

Week 11 [August 13 - August 19]	● Domain specific research. 1. Train a specific vision transformer model for data having multiple patches i.e due to point-like background source. 2. Use augmentation to detect substructure in such strong lenses.
Week 12 [August 20 - August 26]	● Ensemble of models. ● Documentation of the codebase. ● Demo Notebooks of all the implementations. ● If time allows, try to compress the trained models.
Week 13 [August 27 - Sept 2]	● Write reports. ● Buffer time to remove any blockers. ● Any further enhancements.
Final Evaluation [August 28 - Sept 4]	

End-Term evaluation

Goals: Wrap up

By this time, I will ensure that all of the above implementations have been documented and tested. I will extend this period to my **Future Work** by writing blogs explaining all the works and preparing for a major release.

I will update my progress weekly to my mentors and incorporate their feedback and suggestions as and when required. I am confident enough to complete the project on time. I have also kept a buffer week before final evaluations for any critical blockers.

Deliverables

- Accurate deep vision transformer algorithm to identify substructure in strong gravitational lensing images in data sets (both real and simulated) and to do anomaly detection.
- New results on most recent DeepLense datasets for multi-class classification, regression and anomaly detection.
- A GitHub Repo for all the above implementations.
- Beginner-friendly readme for easy installation with detailed steps.

Other Deliverables

- **Contribution to DeepLense:** Apart from the deliverables mentioned above, I would also get more involved in the DeepLense ecosystem by contributing to it and helping other contributors.
- Non-coding tasks: There are several miscellaneous non-code tasks that I would like to take up to give back to the community , such as mentoring.
- Adding more new technologies, improving the tool with innovations into the project.

Future Scope

The Lenses detected by the resulting algorithm can then be passed on to the rest of the DeepLense pipeline for detection, classification and interpretation of dark matter substructure.
I wish to remain as an active maintainer of the DeepLense community. I will also address all the future improvements of the algorithm.

The Motivation for GSoC

I am a typical geek who loves programming and enjoys problem-solving and making side projects a part of hobby coding. Along with my friends, I manage a university-level open-source community, CyberLabs, where I have worked on various open-source projects.

The pride in the feeling that my code will cause an impact in the lives of millions of people who will use it is unparalleled. Moreover, it allows me to grow as an individual and learn how to work in an enormous community team. I have also been active in introducing people to the world of open source and getting them involved with various open-source projects and communities.

Achievements:

1. Received Kaggle Expert Tag in Competitions and Notebooks section.
2. Secured 2nd position in Takshak 2021 - hackathon by IIT Dhanbad.
3. Secured 4th Rank in R-Street Quant Challenge, Kaggle.
4. Selected in top 50 among 20,000 participants in Samsung's Solve for Tomorrow.
5. Secured Silver Medal (Top 10%) in Ubiquant Market Prediction, Kaggle.
6. Secured Bronze Medal in G2Net Detecting Gravitational Waves, Kaggle.

THANK YOU