GSoC 2023 Project Proposal

Organization: ML4SCI

Deep Regression Techniques for Decoding Dark Matter with Strong Gravitational Lensing

MENTORS

Michael Toomey   Stephon Alexander   Brandon Ames   Sanaz Kiyadeh
Yurii Halychanskyi   Karthik Sachdev

AUTHOR

SHUBHAM MISHRA

Contents

1. Introduction and Student Information

- Name: Shubham Mishra
- Email: shubhammishra5658@gmail.com
- GitHub: https://github.com/sam5658
- LinkedIn: https://www.linkedin.com/in/shubham-mishra-52529b226/
- Contact no: +91-8770867683 / +91-7987878037
- University: Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal, India
- Institute: LNCT Group of Colleges, Bhopal, India
- Time Zone: IST

2. Abstract

Gravitational Lensing is a concept of *Matter Distribution* between the source (Galaxy Clusters) and the observer, Sometimes when there is a heavy object with just enough mass (generally a Massive Galactic Cluster or Black Hole) between the source and the observer, light coming from the source gets bend and Multiple Arcs are formed. If the mass of that object is sufficient enough It forms an Einstein Ring. The study of such "strong Gravitational Lensing" images can help us understand the underlying nature of Dark Matter.

DeepLense utilizes the potential of Deep Learning techniques and accurately predicts and differentiates the Substructures of Dark matter. This project aims to explore various 'Deep Regression models to predict the dark matter substructure's properties. The Project also aims to deliver tutorials and basic scripts to train and use the Potential Models.

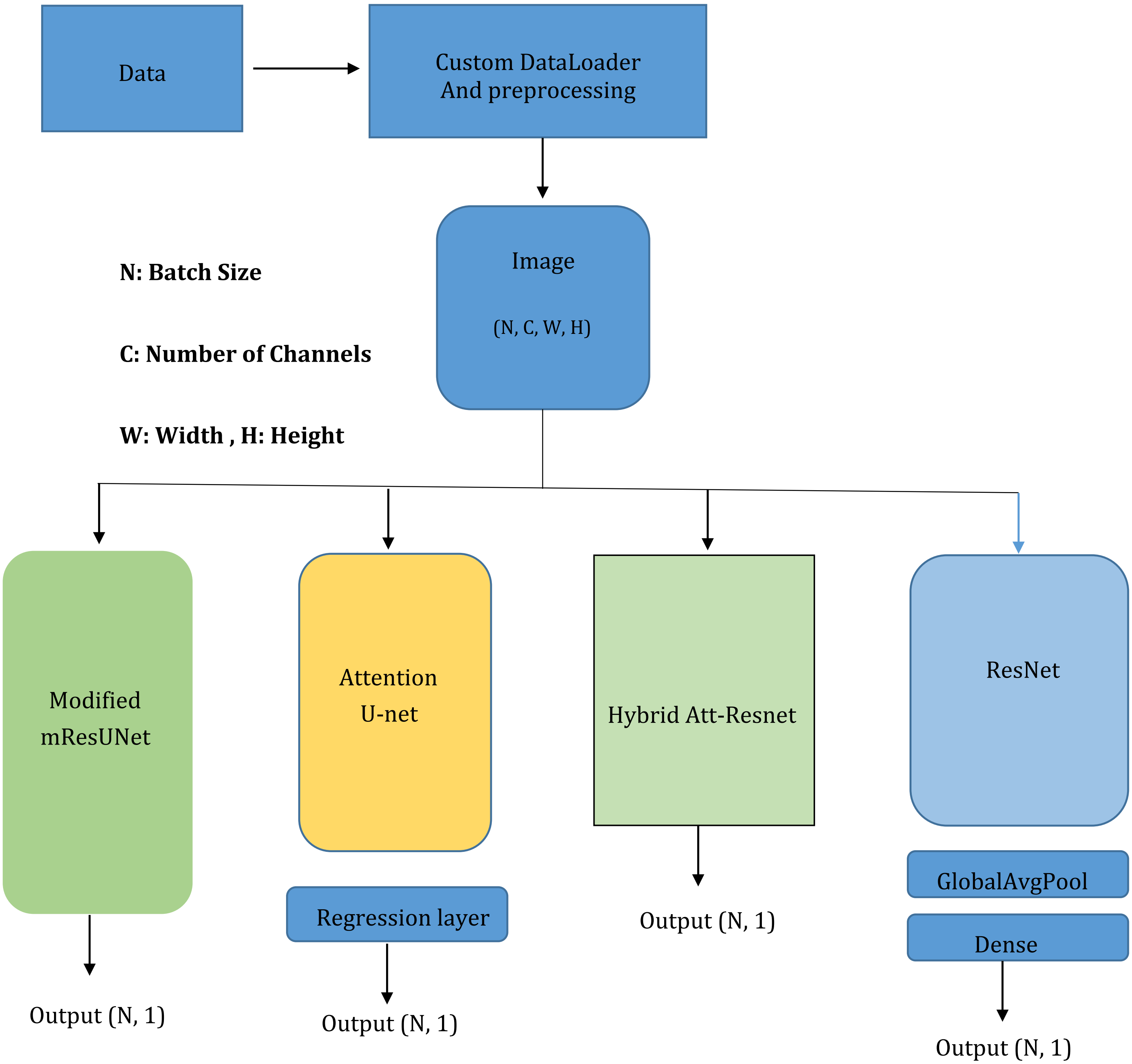
3. Project Description

DeepLense is a cutting-edge Machine Learning (ML) *pipeline that facilitates a comprehensive understanding of various properties associated with dark matter particle candidates, including but not limited to cold dark matter (CDM), warm dark matter (WDM), axions, and self-interacting dark matter (SIDM)*. Dark matter is a crucial component of the universe, and its elusive nature has made it a subject of intense research in the field of astrophysics. The DeepLense project focuses on leveraging advanced DL techniques to conduct regression analyses and classify and estimate the properties of Dark Matter substructures, such as axions, vortex and sphere.

The project employs Deep regression techniques [1] that involve training layers of convolutional neural networks, followed by feed-forward neural networks. These architectures are commonly known as vanilla deep regression models. The project's primary goal is to regress dark matter properties (such as the total fraction of axion or vortex mass present in the Dark Matter) with the help of deep regression models suggested in figure 3.1. The models are selected with the idea of combining attention modules with deep regression models.

3.1 Proposed Models

This section shows the workflow and models that I plan to use for the regression task.


Data
Custom DataLoader
And preprocessing
Image
(N, C, W, H)
N: Batch Size
C: Number of Channels
W: Width , H: Height
Modified
mResUNet
Output (N, 1)
Attention
U-net
Regression layer
Output (N, 1)
Hybrid Att-Resnet
Output (N, 1)
ResNet
GlobalAvgPool
Dense
Output (N, 1)

3.2 Model Description

1. MResUnet :

The mResUnet model improves the U-net model specifically for regression tasks by adding multiple residual neural networks to it. The model I plan to use is inspired by the paper Mass Estimation of Galaxy Clusters with Deep Learning I: Sunyaev-Zel'dovich Effect [2]

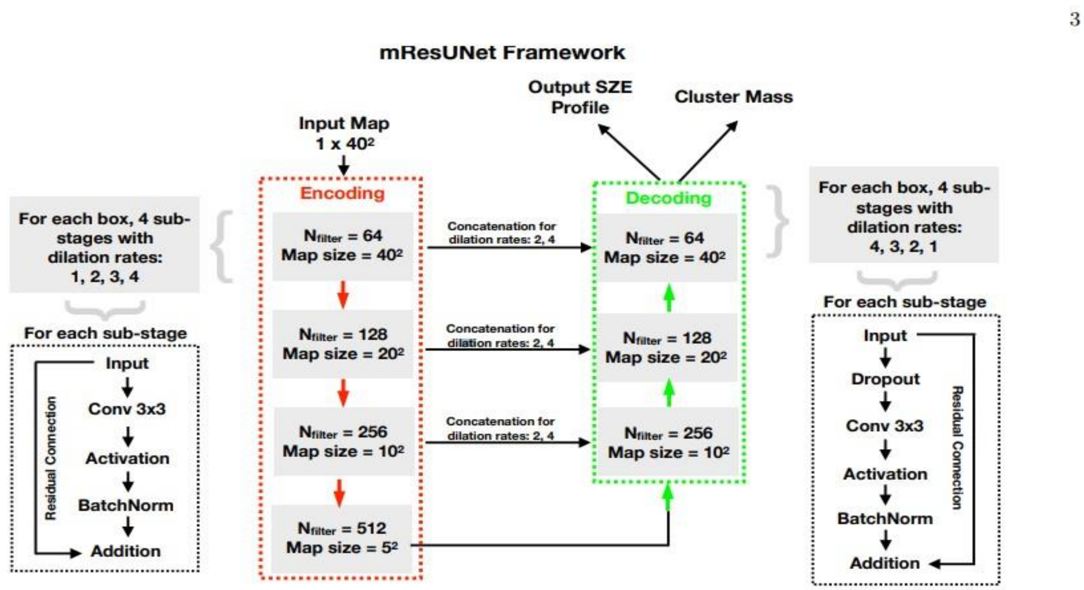


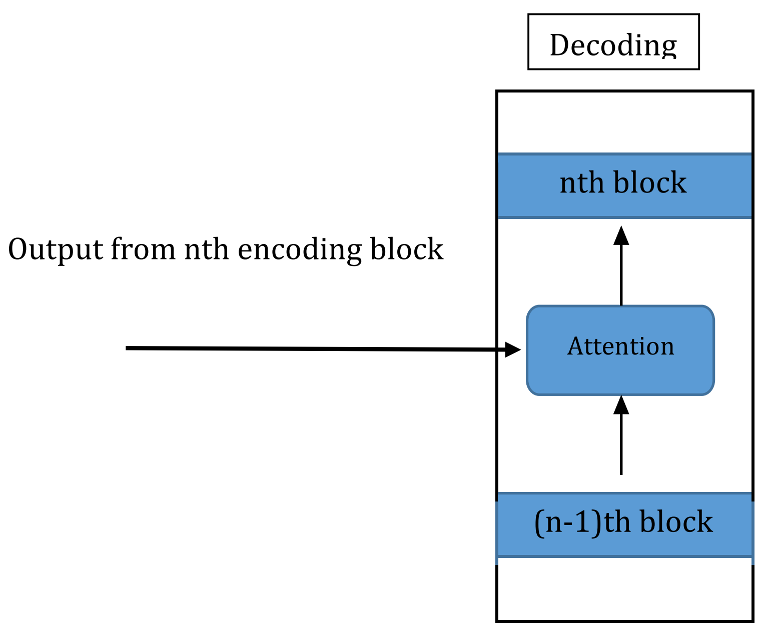
Figure 1. The mResUNet framework with decoding (red dashed box) and encoding phases (green dashed box). Each gray coloured box in these phases represents a convolution block. We change the number of filters and the map size by down sampling (red arrows) and up sampling (green arrows) the feature maps in the encoding and the decoding phases, respectively. The convolution block has four sub-stages where convolution operations are applied with different dilation rates of N = 1, 2, 3 and 4. All sub-stages have convolution, activation and batch normalization layers, and residual connections are applied between the input and output feature maps. The sub-stages of convolution blocks in decoding phase have an extra dropout layer to prevent model over-fitting. Skip connections are used to concatenate feature maps from the encoding convolution blocks to corresponding blocks in decoding phase that helps in retrieving the lost spatial information due to down sampling (see Section 2.1).

Figure: 3.1 mResUNet Framework [2]

Implementation:

The mresunet model as shown in the figure can easily cause overfitting due to a large number of parameters, thus I propose using modified mresunet, in this modified version I will first train the basic mresunet model and will later try to decrease or increase the dilation rates and decoding stages and will finally implement the best model.

There is one more modification that I plan to do, which is to use the attention module instead of normal addition skip connections. Attention is a pretty good approach for keeping the relevant features, it is a basic concept for giving more attention to a specific area or feature. There are good ways of implementing a self-attention Module. I will first try to use a basic attention module and if it looks promising I'll also try to use the attention gate proposed in Attention U-net which is shown in figure 3.2. The decoding section after adding the attention block to mresunet will look as follows.



2. Attention U-Net

The attention U-net Model is inspired by the paper Attention U-Net: Learning Where to look for the Pancreas [3]. It has the same attention gate that I plan to utilize in mResUNet.

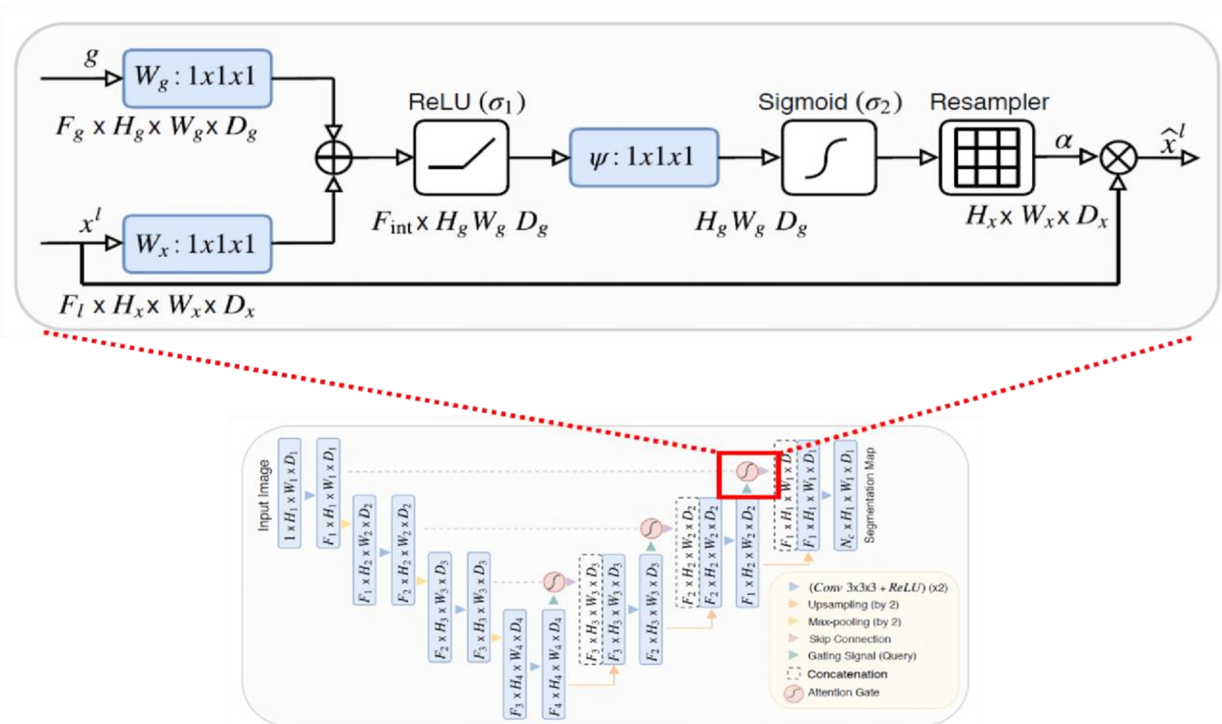


figure 3.2 Attention U-net

The figure shows a block diagram of the proposed Attention U-Net segmentation model. The input image is progressively filtered and down sampled by factor of 2 at each scale in the encoding part of the network (e.g. H4 = H1/8). Nc denotes the number of classes. Attention gates (AGs) filter the features propagated through the skip connections. Latter the pooling and Flatten layer can be implemented, for achieving better results on our given datasets, further dense layers can be added to the network to finally give the output of shape (number of batches, 1).

3. Hybrid Att-Resnet

The attention resnet model is explained in the paper attention resnet [4] and involves several attention modules followed by pooling and feedforward networks, here we can utilize the architecture by adding the regression layer for performing the regression task. Such models are considered for deep regression for not having large dense resnet networks but leveraging attention modules with resnets.

4. Resnet

Resnets are one of the potential deep learning neural networks which were formed for keeping the features of input map for longer layers, I will try various resents(18, 34, 50, 101) for performing regression, also resent-50 is the exact model that I used in evaluation test for regression task, The resnets have the following architecture:

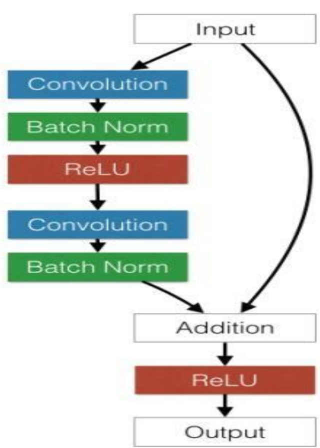
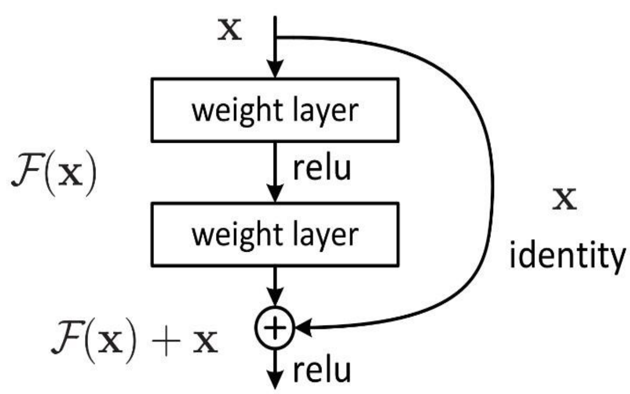


Figure 1. A RestNet basic block



$$Y = F(x) + x$$

3.3 Criterion and Metrics

- The optimizers I plan to use are Adam, Ranger or RAdam. There will be more clarification on what optimizers to use after studying the dataset and training the models and implement the best in my final Product.
- For Loss functions I will be using MSE (Mean Squared Error), RMSE (Root Mean Squared error) and MAE (Mean Absolute error).

Implementation

Models mentioned above in 3.2 have a few modifications as explained for the sake of performing regression like adding a regression layer, changing the input channels, adding attention modules etc.

If there's any need to code anything from scratch I can do it, I will prefer implementing everything using pytorch, but there are a few things like Ranger optimizer which can be more easily accessed with the help of pytorch libraries like fastai, thus if needed I can also use other libraries like fastai or code things from scratch with pytorch.

4. Deliverables

- Using the promised Models for performing regression for estimating the dark matter substructure’s properties.
- Documenting the progress
- Creating tutorials with detailed explanations for training and using the Models
- Creating basic scripts for training, testing and using the files.

4.1 Schedule

Community Bonding Period May 4 – 28

- I will start bonding with the mentors and engage with the community.
- I will start with reading and understanding the important documentation and blogs.
- I will start engaging with the community and discuss the dataset and will start contemplating on required preprocessing.

Coding officially begins! May 29 – June 12

- I will start developing mResUnet model. I will try to further study and improve the model for minimizing the loss.
- Before this, I will be creating the custom dataset and Data Loaders which will include all the preprocessing and loading datasets.
- I will also try to discuss further scopes of improvement with the mentors.

June 13 – June 19

- By utilizing the methodology outlined in the preceding section 3.2 incorporating the Attention U-net architecture, which entails the inclusion of a regression layer, followed by additional fine-tuning procedures to optimize the model's performance, effectively enhancing effectiveness of our implementation.

June 19 – June 29

- During this particular session, my primary focus and objective will be on incorporating and utilizing the attention module, which will involve the integration of this feature into the mresunet model. By doing so, we hope to gain a more thorough understanding of the attention module's capabilities and potential for enhancing the performance of our model.

Phase-1 Evaluation - June 29 – July 14

- This duration will be a crucial one as it involves documenting my work, creating blogs etc. I will finish all the work that's been left due to some delay and will submit the result for phase one evaluation.

July 14 – July 25

- Developing a Hybrid Attention resnet as explained. Experimenting with different hyperparameters and criterions for performing regression.

July 25 – Aug 6

- Finally implementing the Resnet Model, here I will train a few resnet models such as resnet-50. Adding further layers for regression and fine-tuning the model.

Submission of the Final Product - Aug 6 – Aug 21

- In the final GSoC phase, I will be creating notebooks for tutorials, developing scripts for navigating, training and using files in repository.
- Publishing the blog Posts.
- The Final coding period will involve testing and further tuning of the models if necessary
- Getting further suggestions from the mentors, I'll review my work and will submit my Final product for the phase – 2 evaluation.

4.2 Additional Information:

- I will be available for working on the project from 6:00 pm IST. (I have my classes from 10:30 am – 5:00 pm) during the weekends I can work for any good hours.
- In a week I can work for 30 – 40 hours.
- I will be having my End Semester exams around mid-June and I will inform my mentors a month before If in case I have to take any sought of leave.

5. Motivation

My Motivation for contributing to the organization ML4SCI stems from my deep-rooted desire to contribute something to the astrophysics community. As a huge….. Lifelong fan of physics being spontaneously attracted to DeepLense Projects was pretty obvious (because what will be cooler than working with the dark matter related projects). I've developed great enthusiasm towards deep learning starting from my first year at college. My long-term career goal is to develop more skills and use them to contribute to the physics community, as being in 2nd year of my degree there weren't many opportunities for physics enthusiasts like me and that's when I thought going open source, thus I got the idea of participating in GSoC. I explored the projects last year but it was a bit late so I decided to contribute next year In fact, I'm not even applying to any other organizations. I'm really excited and overwhelmed by the idea of working with DeepLense. I'm really interested and passionate about working on things which align with my interest.

I will be highly grateful if you could provide me with this great opportunity of contributing to DeepLense as a Google Summer of Code student and will try to work to the best of my abilities.

Good Fit

I've all the skills required for the completion of the proposed work. I know the importance of working in a community and focusing on writing efficient and readable code. I highly commit to the schedule and will finish all the promised work before the deadline. I've been majorly known among my peers for bonding and communication. I am a great enthusiast of deep learning I started with machine learning and deep learning in my 1st year of college and have been focusing on developing more skills and aim to make a career where I can use my skills and abilities to develop solutions to difficult scientific problems. I also have sound knowledge of data structures and algorithms in C++, the academic classes (till the 3rd semester) that I took included calculus and graph theory, thus giving me a pretty good mathematical background. I will regularly be in touch with the mentors and will discuss my plans and problems with them. I am proficient in speaking Hindi and English.

I promise to provide all the proposed deliverables as per the schedule. The project will correspond to a duration of 350 hours in case of any leave or minor changes in schedule I will try to update the mentors at least a week before.

You can see my resume for further details.

DeepLense and Evaluation Task

I have a comprehensive understanding of all the deep learning concepts, I am a fast learner, and In the process of learning deep learning, I have developed a strong habit of coding models from scratch. PyTorch is my preferred framework. I've read previous DeepLense blogs [5] and have grown familiar with concepts like strong gravitational lensing, dark matter substructures etc. I have cloned the ML4SCI DeepLense repository multiple times and have tried to understand the available projects in more detail, thus I know the coding style used in previous year's projects.

The Evaluation Tasks can be found here:
GitHub

6. References

[1] Deep Learning The Morphology of Dark Matter Substructure

[2] Mass Estimation of Galaxy Clusters with Deep Learning. I. Sunyaev–Zel'dovich Effect

[3] Attention U-Net: Learning Where to Look for the Pancreas

[4] Residual Attention Network

[5] GSOC 2021 with ML4SCI | Deep Regression for Exploring Dark Matter