Google Summer of Code

Gravitational Lens Finding for Dark Matter Substructure Pipeline

ML 4 SCI

About Me

- Name: Mriganka Nath
- GitHub: https://github.com/mrinath123
- University: National Institute of Technology, Silchar
- Time zone: Indian Standard Time (UTC +5:30)
- Email Id: mrinath123@gmail.com

Project Description

Background

Dark matter is one of the most mysterious elements of the cosmos; it is estimated to account for around 85% of all matter in the universe. Dark matter is extremely difficult to detect. Researchers are still working hard to figure out what dark matter is and what its characteristics are. Calculations show that many galaxies would behave quite differently if they didn't include a large quantity of unseen stuff, which serves as the major evidence of dark matter. Given the scale of experimental efforts aiming at identifying dark matter, clues to its nature have proven unexpectedly elusive. While terrestrial tests may be able to nail down a model, another possible option is to discover dark matter based on astrophysical or cosmic data. A particularly sensitive approach is based on the unique signature of dark matter substructure on strong lensing images.

When a massive volume of matter, such as a cluster of galaxies, forms a gravitational field that distorts and magnifies light from distant galaxies that are behind it but in the same line of sight, this is known as a gravitational lens. When there is a single concentration of matter at the centre, such as the dense core of a galaxy, gravitational lensing is the simplest kind. Observations of huge clusters of galaxies reveal more complex gravitational lensing. Background galaxies are distorted by the cluster, and their images may be seen as short, thin "lensed arcs" at the cluster's edges. These distorted pictures also serve as probes for the galaxy cluster's matter distribution. The findings show that most of the mass in a galaxy cluster is neither visible galaxy or hot gas around them, and so is called dark matter.

Strong lensing has been effectively used to investigate dark matter substructures. It's even been utilised to differentiate between different kinds of dark matter substructures. Deep Learning models may be able to differentiate between different forms of dark matter substructures.

Motivation

Machine learning techniques are regarded to have the potential to help researchers better comprehend dark matter. Convolutional Neural Networks (CNNs) have previously been used to successfully distinguish between distinct types of substructures such as no substructure, vortex substructure, and spherical substructure [1]. Using strong lensing simulations, unsupervised learning has also been shown to aid researchers in inferring the presence of substructure in dark matter halos [2]. In this task, we'll classify images based on whether or not they have a strong lensing effect. Strong Gravitational Lensing allows us to analyse the substructure of dark matter, allowing us to gain a deeper understanding of its underlying nature. The observed strong lens images may be utilized in the Deeplense pipeline for dark matter substructure classification, anomaly detection, and interpretation. Images may be processed using a variety of algorithms. We may experiment with other types of CNNs as well as transfer learning. We are provided several features in the data, such as the Einstein area and total flux in the picture pixels emanating from the lensed source alone in the R band, in addition to the images. To improve our findings, we can have our models use both image data and "meta" features.

Project overview

Abstract:

The project's goal is to develop machine learning models that can classify images with strong lensing or not on actual and simulated data, which will aid the Deeplens pipeline. We'll try out several types of models, fine-tune them to obtain good outcomes, and compare them. Because a single model may not be sufficient, we can combine the findings of several models using ensembling to obtain a more robust and better outcome. We'll document and record every outcome as we experiment with our models, and we'll refactor the code after the experiment is through.

Key Tasks:

- Conducting literature survey and proper analysis of the data before applying any machine learning model.
- Study different machine learning algorithms and implement them in PyTorch.
- Training and testing these models and compare their performance.
- Integrate the model with a logging library, like WANDB to monitor the performance in real time.
- Investigate if meta features assist the model for better performance.
- Integrating the developed models with the DeepLens pipeline.
- Documenting all the implementation, results, and analysis in detail.

Detailed Project overview:

1) Dataset.py

Python file describe all the pre-processing and augmentations to be done on the data before feeding it to the model.

2) Train.py

A python file to train our models written with Pytorch which will support both CPU and GPU training. Will contain the Loss function, optimizer, learning rate scheduler and necessary code required for training. Could be used directly from terminal.

Input

- The dataset to be trained on.

Output

- Model weights that can be used further for test.
- ROC curve, confusion metrics diagram.

3) Test.py

A python file that will give us the predictions of the trained models on corresponding test images. Could be used directly from terminal and can be tested both in CPU and GPU.

Input

- Trained model weight
- Test images

Output

- Prediction and probability of which class does it belong.
- Class Activation Map for the test image (only if specified by user)

4) Model.py

Will contain all the implementation of the neural network models.

5) Configuration file

A JSON or YAML file containing all hyperparameters in one location, such as the number of epochs, batch size, types of augmentations, learning rate, and so on. Would make it easier for us to conduct our experiments as it will help tidying up the code.

6) Documentation

- List all dependencies, libraries which will be used.
- Release all the trained models.
- Display the evaluation metrics diagrams/results for each model and also draw the architecture of the models used.
- Use a logging tool like Weights and Biases (WANDB) to integrate the model we're training. Will allow us to monitor our model's performance in real time, and will be accessible to all project team.

By carefully consulting with the mentors, I will add more files or merge the aforementioned files into one if necessary.

Proposed Timeline:

March 07 - April 19,2022 (Application Period)

- Study related papers and present a baseline approach for the classification problem and be familiar with the libraries and framework.

April 19 - May 20,2022 (Acceptance Waiting Period)

- Make sure a proper development environment is set up properly with all the necessary dependencies and libraries to work with the data and models.
- Understanding the working of PyAutoLens library.
- Make a to-do list of the tasks that must be completed throughout each phase.

May 20 - June 12,2022 (Community Bonding Period)

- Brush up the necessary concepts required for making the models.
- Discuss on implementation and pre-processing techniques on the data with the mentors.
- Re-iterate the project details with mentors and discuss any workflow details.

June 13,2022 (coding begins)

- Analyse the data and become familiar with the DeepLens pipeline by this time.
- Through a comprehensive literature review, will enlist some pre-trained CNNs architecture that I will be applying.
- Creating some custom CNN models and comparing the results.

July 25-July 29,2022 (First Evaluation)

I plan to complete these tasks by the end of phase 1:

- Finish exploration of the data and find effective pre-processing technique for the data before feeding it to a model as discussed with the mentors.
- Implement selected CNN models and compare the performance.
- Investigate ways on how to combine meta features and image for the prediction.
- Integrate the models with WANDB for real time monitoring of their metrics.
- Document and record every result while doing the above experiments.

September 05-September 12,2022 (Second Evaluation)

By the time of the final evaluation, I plan to do the following tasks:

- Find the relation and effect of features on the model's prediction.
- Ensemble the predictions of CNN models to see if it gives better result or not.
- Release the weights of our trained models, so that anyone can use these models further in the DeepLense pipeline.
- Refine my existing documentation and provide starter notebooks with proper comments describing how to train and test the models so that it will be easy for anyone to understand our work.
- Submit the final work

September 20,2022 (Results Announced)

Deliverables

- Pre-trained model weights for all the model we experimented on and can be directly used further in the pipeline.
- Starter notebooks with appropriate comments outlining how to train and test the models.
- Class activation Maps for the predictions to understand what the model has learned and where in the image it is focusing.
- Display related evaluation metrics diagrams/results, plots, graphics and also draw the architecture of the models used.

Other Commitments

I am in my last year of my bachelor's degree, and my end-of-semester exam will be completed by the second week of May 2022, therefore I will be totally devoted to the project following the exam. For the length of the project, I intend to devote 7-8 hours every weekday and 3-4 hours per weekend. Furthermore, ML4Sci is the sole organisation for which I am applying.

Relevant Experience

- I have worked on projects and won Kaggle competitions on solutions written in python and pytorch.
- I was ranked 3rd in the Particle classification challenge in the ML4SCI 2021 hackathon.
- Worked with the Pytorch vision github repository and fixed an example given in their documentation.[Merged]

References

[1] Alexander, Stephon, Sergei Gleyzer, Evan McDonough, Michael W. Toomey, and Emanuele Usai. "Deep Learning the Morphology of Dark Matter Substructure." The Astrophysical Journal 893, no. 1 (2020): 15.

[2] Alexander, Stephon, Sergei Gleyzer, Hanna Parul, Pranath Reddy, Michael W. Toomey, Emanuele Usai, and Ryker Von Klar. "Decoding Dark Matter Substructure without Supervision." *arXiv preprint arXiv:2008.12731* (2020).