ML4SCI

Google Summer of Code Project Proposal

Self-Supervised Learning for Strong Gravitational Lensing

Contact Information :

Full Name : Om Anil Doiphode

University: Veermata Jijabai Technological Institute (VJTI) , Mumbai

Course: Bachelor of Technology (BTECH)

Branch: Computer Engineering

Email: omdoiphode161@gmail.com

oadoiphode_b21@ce.vjti.ac.in

Physical Location: Mumbai, Maharashtra ,India

Time Zone: Indian Standard Time (UTC+5.30)

GitHub

LinkedIn

Resume

To give feedback, please contact the email address.

Content

1. Overview

1.1 Project Abstract

- Strong gravitational lensing is a useful tool to study the substructure of dark matter and its nature.
- Deep learning methods can help identify images with substructure and differentiate between different dark matter models.
- Supervised classification can be challenging when there are only a few samples of a particular class.
- Self-supervised learning has been successful in cases where labeled data is limited.
- Convolutional neural networks have been used for self-supervised learning in strong gravitational lensing data, but Transformers or hybrid models have not been explored.
- This project aims to develop self-supervised learning techniques with Transformers for strong gravitational lensing data.
- The project will also explore equivariant techniques in the self-supervised learning context for other strong lensing tasks.
- EsViT model will be implemented during this project which is self supervised equivariant transformer.

1.2 Benefits to Community

Studying the substructure of dark matter using strong gravitational lensing and deep learning methods has several potential benefits:

1. **Understanding the nature of dark matter**: Dark matter is one of the most fundamental mysteries of the universe. By studying its substructure, we can gain insights into its properties and help narrow down the list of potential candidates for what it might be.
2. **Testing different dark matter models:** There are several well-motivated models of dark matter, including WIMPs, axions, and axion-like particles, as well as warm dark matter. By using deep learning to differentiate between these models based on their substructure, we can test which models are more consistent with observations.
3. **Advancing our understanding of gravitational lensing:** Strong gravitational lensing is a powerful tool for studying the structure of the universe. By improving our ability to identify substructure in lensing images, we can improve our

understanding of how gravity works on a large scale and how galaxies and clusters of galaxies form.

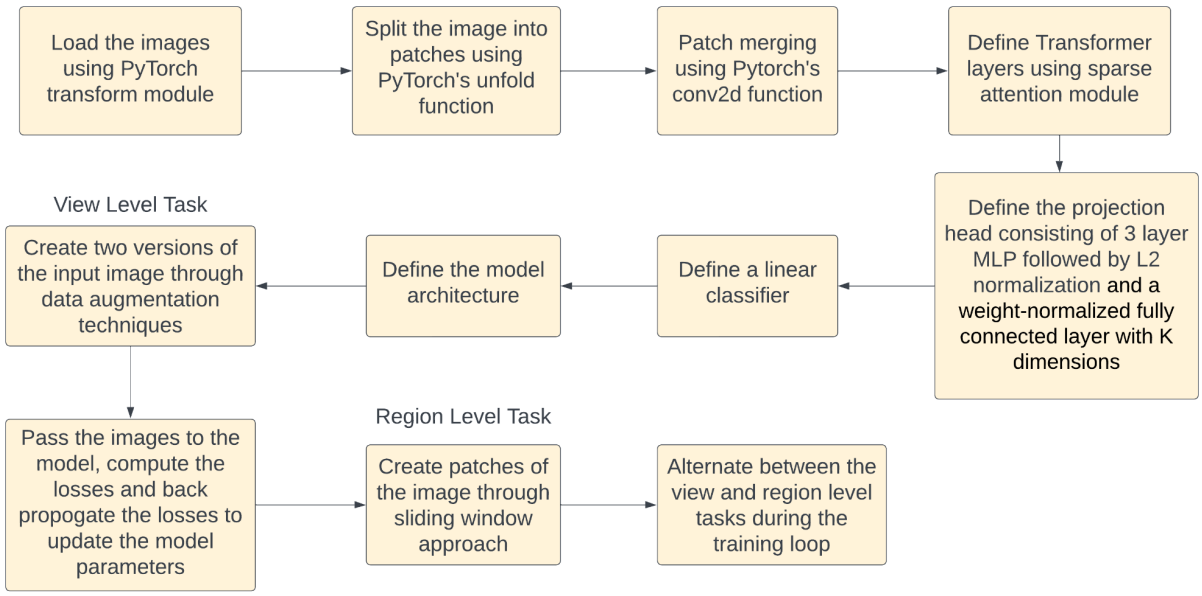
4. **Developing new machine learning techniques:** The project also has the potential to drive the development of new machine learning techniques that can be applied to other areas of astrophysics and beyond. Deep learning has already shown great promise in a variety of fields, and this project could help push the boundaries of what is possible.

2. Goals

- Explore the use of **Equivariant Transformers with self-supervised learning for representation learning**. The trained model could then be fine-tuned for specific tasks such as regression or classification.
- Expand the **DeepLense functionality** with **self-supervised learning algorithms** suitable for **computer vision** tasks applicable to strong gravitational lensing data.

3. In Depth Description

- Self-supervised learning with Transformers in NLP has achieved state-of-the-art performance in various tasks.
- The approach trains a language model to predict missing words or sentences in a text corpus without using explicit annotations or labels.
- The pre-training process enables the model to learn meaningful and contextualized representations of words and sentences that can be fine-tuned for downstream NLP tasks.
- In computer vision, CNNs have been the dominant architecture for self-supervised visual representation learning.
- Recent works have shown that self-supervised learning with Transformers can be effective in computer vision tasks and can outperform CNN-based approaches.
- Transformers can learn more structured and interpretable representations than CNNs, which can be beneficial in tasks such as object detection and semantic segmentation.
- The attention mechanism in Transformers allows the model to capture long-range dependencies and contextual information that might be missed by CNNs.



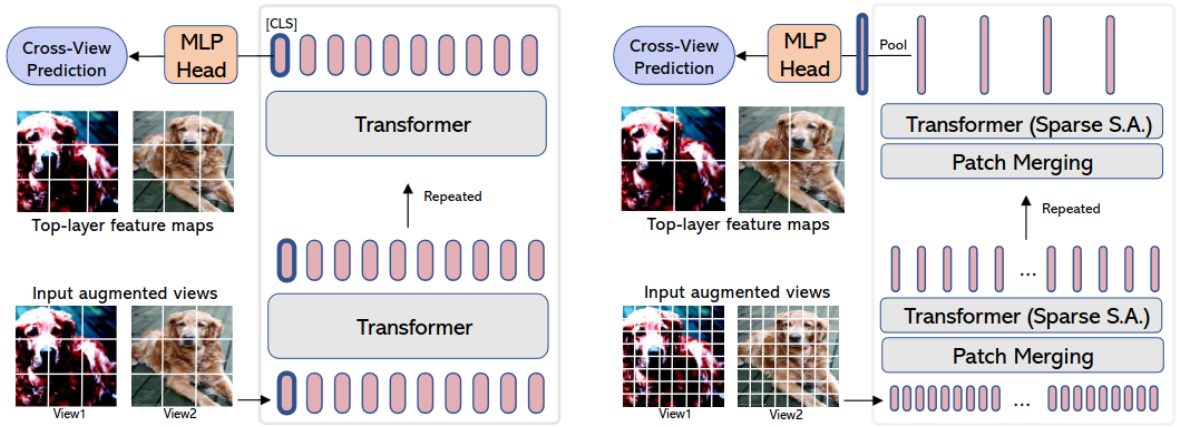
Working of the proposed approach

3.1 Self supervised learning through multi staged transformer and region based pre training task.

(1) Traditional self supervised learning monolithic transformers automatically discover semantic correspondences between local regions.

(2) However they are computationally expensive and therefore a multi staged transformer model is being considered. Though computational complexity is reduced, the multi stage architecture results in loss in correspondence property.

(3) A region based pre training task can be employed to alleviate the above issue.



(a) Baseline monolithic architecture (b) Multi-stage architecture

- The model consists of several stages, each including a patch merging/embedding module and a Transformer with sparse self-attention module.
- The patch merging module splits an RGB image into non-overlapping patches and treats each patch as a "token," which is further projected into a feature vector.
- In later stages, the patch merging module concatenates neighboring patch features and applies a linear layer to reduce tokens and increase output dimension.
- The Transformer with sparse self-attention module enables interactions among merged features to generate hierarchical representation.
- We can implement such a model either from scratch or by using a pretrained model. Implementing such a model from scratch might take a significant amount of time so using a pretrained model like **Swin Transformer** can be a feasible option. Swin Transformer incorporates both patch merging and sparse self attention property. Further decisions with mentors will help in deciding which approach to take.

Self Supervision in DINO transformer

- A non-contrastive framework is employed to build SSL method based on Self-distillation with no labels (DINO) approach.
- DINO uses knowledge distillation, where a student network is trained to match the output of a teacher network.
- Both networks have a backbone and a projection head, and the features used in downstream tasks are the output of the backbone.
- Different augmented views of an image are fed into the backbone network to obtain feature maps.
- Feature maps are converted into probability vectors by two MLP heads followed by softmax for view-level and region-level prediction.
- DINO does not rely on contrastive learning and incorporates equivariance in the network.
- Comparing representations of different views of the same image helps identify the underlying structure invariant to transformations, improving representation quality and equivariance of the Transformer model.

View-level task

- A view-level task involves evaluating a computer vision model on a set of augmented views of images.
- A set of pairs is constructed to perform cross-view prediction tasks.

- The pairs are selected from corresponding augmented views in the student and teacher models such that the original images are the same.
- The model is evaluated on its ability to predict the same output for both the student and teacher views.
- The objective is to minimize the difference between the predicted probabilities of the student model and the teacher model for each corresponding pair of views.
- The loss function used to measure this difference is MV(s,t) = -ps*log(pt), where ps and pt are the probability outputs of the MLP head h for the student and teacher view-level representations, respectively.

$$\mathcal{L}_V = \frac{1}{|\mathcal{P}|} \sum_{(s,t)\in\mathcal{P}} \mathcal{M}_V(s,t), \quad \text{with} \quad \mathcal{M}_V(s,t) = -p_s \log p_t,$$

Lv Loss Function

Region-level task

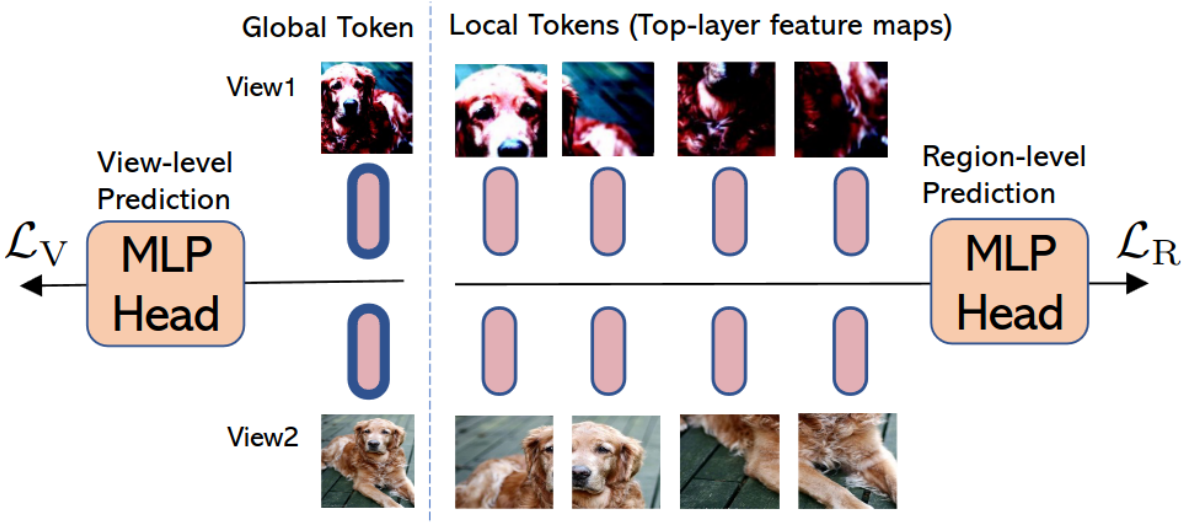
- The LV loss function encourages local-to-global correspondences between large and small crops of an image, but it does not enforce region-to-region correspondence between specific regions of an image.
- A region-level task can be used to explicitly enforce region-to-region correspondence between the student and teacher models, using a contrastive loss function.
- This region-level task aims to learn features related not only across different scales of an image but also between specific regions of the image.
- Region-level pre-training task can amortize computation, fully leverage extracted region-level features, and take into account co-occurrences/structures between local features.
- Directly performing masked patch prediction (MPP) for multi-stage Transformer architecture is not feasible because the correspondences between input visual tokens and output features get diluted due to the merging operation.
- Even for monolithic architectures, MPP has not been proven effective in computer vision.

To address this problem, a non-contrastive, region-matching method is used that directly works at the level of local features by taking into account their correspondences:

$$\mathcal{L}_R = \frac{1}{|\mathcal{P}|} \sum_{(s,t)\in\mathcal{P}} \mathcal{M}_R(s,t), \quad \text{with } \mathcal{M}_R(s,t) = -\frac{1}{T}\sum_{i=1}^{T} p_{j^*} \log p_i, \quad j^* = \arg\max_j \frac{z_i^T z_j}{\|z_i\|\|z_j\|},$$

where $p_i = h'(z_i)$ and $p_j = h'(z_j)$ are the probability outputs of a new MLP head h' over the local features of student $z_i \in z_s$ and teacher $z_j \in z_t$, respectively. $j*$ is the index of the feature in z_t that best matches the i-th feature in z_s, in the sense of highest cosine similarity. Note that z_i and z_{j*} are contextualized features of two best matched regions from different augmented views, minimizing L_R encourages different contexts (i.e., surrounding regions) to learn invariant features, and thus captures the region-dependency.

- The EsViT pre-training objective includes both view-level (LV) and region-level (LR) prediction tasks.
- The overall objective is the sum of the two objectives: L = LV + LR.
- The student network gθs is updated iteratively while the teacher network is updated as an exponential moving average (EMA) of the student network's weights.
- The updates are performed by minimizing the full cross-entropy loss for the student network and updating the teacher network as an EMA of the student weights.
- The update process is repeated iteratively until convergence, with λ following a cosine schedule from 0.996 to 1 during training.



Pre-training objectives, including view-level (left) and region-level (right) prediction.

3.2 Approach followed in the evaluation task

In the evaluation task, we were supposed to build a self supervised transformer for binary classification. For that task I have used BEiT which is also a self supervised transformer model from hugging face. It doesn't incorporate equivariance but we can incorporate it using the approach discussed in section 3.1.

More details can be found here: Specific test

4. Timeline and Research Plan

Period	Work Expected
4.1 Application Review Period (20th March- 4th May)(On my mark)	• Getting familiar with different transformer models like ViT (Vision Transformer), Swin, BEiT and EsViT specifically designed for computer vision tasks. • Learning about the different self supervision techniques currently being used to get an idea on how to combine self supervision with transformer architecture. • Researching more about the concepts discussed above to get a better understanding of the project implementation.
4.2 Community Bonding Period (4th May - 28th May)(Get Set)	
Week 1 (4th May - 10th May) : Planning	• Getting feedback from the mentors about the submitted proposal. • Discussing with mentors the high level details of the project (goals, priority and deliverables, meeting schedule). • After this, create a general structure of workflow of the project and adjust timeline

	and research plan accordingly.
Week 2 (10th May - 17th May): Getting things ready	• Prepare the working environment and repository. • Obtain and clean the dataset. • Do all necessary preprocessing (normalization) in the Jupyter Notebook.
Week 3(17th May- 28th May): Bonding	• I would like to hear other students' as well as mentors' perspectives on the use of deep learning for dark matter prediction and how self supervised learning can aid in computer vision tasks applicable to strong gravitational lensing.
4.3 Coding (29th May- 28th August)(GO)	
Week 1-2 (29th May- 11th June)	• Code and train EsViT model from Microsoft in Jupyter Notebook. • Do the training several times, save the performance statistics and the models. • These performance statistics will be used as a benchmark when developing the Equivariant Transformer with self supervision learning. • Currently only EsViT is an equivariant transformer model with self supervision. • Make required changes to the first approach report and notebook as per feedback.
Week 3-8(12th June - 24th July)	• Build Multi staged transformer starting from building a monolithic transformer and then augmenting it with additional layers expected in a multi stage architecture. • Though straightforward in implementation,

	without careful treatments, we may lose some useful properties of monolithic model. So 1 week will be utilized in developing an effective model. ● In the following 3 weeks, develop view level and region level tasks along with the help of mentors required for self supervised learning and also to incorporate the property exhibited by monolithic transformers. ● Develop other self supervised algorithms suitable for computer vision tasks applicable for strong gravitational lensing. ● Submit a short evaluation to the mentors in the second last week of this part of the project. **(Week 7)**
Week 9-10(25th July- 8th August)	● Benchmark the built equivariant transformer model against existing self supervised transformer models. ● Visualize the comparison, e.g., the training and testing loss curve, the training and testing accuracy curve, and the final models' ROC curve and AUC score. Tabulate the average accuracy and AUC score of the best model from both our model and classical methods. ● Note down the performance comparison analysis from the visualizations.This analysis will be reused as the final report's material.
Week 11(9th August - 15th August)	● Clean up the Jupyter Notebook for training. ● Prepare the documentation.
4.4 Students Submit Code and Final Evaluations (24 Aug- 4 Sept)	● Create the final evaluation and submit to mentors.

5. About Me

I am currently a sophomore at Veermata Jijabai Institute of Technology, Mumbai pursuing Bachelor of Technology in Computer Engineering and will graduate in June 2025. During my high school days, I was highly interested in astronomical events, used to read blog posts, articles related to advancements in research in this enigmatic field. The concept of dark matter has been fascinating to me because it challenges our understanding of the universe and the laws of physics. It suggests that there is much more to the universe than what we can directly observe with our current technology and that there is still much to be discovered.

I have experience in Python, C, C++, React, Node. I have also made projects in Computer Vision and Deep Learning.

I don't know if it's relevant to say here but one thing about me is that if some error or problem occurs, I try to find the solution then and there even if it takes hours to get resolved or may not get resolved at all. I am highly passionate about what I do and always try to give my 100% to the job in hand.

6. Open Source Contributions

1. **DeepForest**
   Fixed deprecation warning when running main.deepforest()
   https://github.com/weecology/DeepForest/pull/375 (merged)

2. **Ivy**
   - **Add logical_and method to PyTorch Frontend**
     https://github.com/unifyai/ivy/pull/13114 (merged)
   - **Add bitwise_left_shift method to PyTorch Frontend**
     https://github.com/unifyai/ivy/pull/13120 (merged)
   - **Add Linear Algebra operations to PyTorch Frontend**
     https://github.com/unifyai/ivy/pull/13162 (open)

   - **Add bitwise_right_shift method to PyTorch Frontend**
     https://github.com/unifyai/ivy/pull/13116 (open)

After contributing to open source for quite a long time I have a good understanding of **git**, **github workflows** and the version control system.

7. Other Commitments

I have my summer holidays during June to July and won't have any major commitments. I will devote **8-9** hours daily during **June to July** which I would divide into two parts: 7-8 hours for the actual project and 1-2 hours(learning time) and **3-4** hours daily during **August to September** considering my college hours.

However, I will do my best not limiting myself to work only for a few hours per day but continue working as long as I can and get the best out of this opportunity.

Considering the small amount of time allotted to this project, I would like to continue working on this project even after the end of the program(if the organization allows it) and keep contributing to this field.

Note: I have considered extended time durations on a daily basis because there can be unprecedented errors, system issues which may slow down the project work. The learning time mentioned is considered because I want this project to be a learning experience rather than just completing it as a task.

8. References

[1] EsViT Paper

[2] BEiT Paper

[3] DINO Transformer Paper

[3] https://github.com/microsoft/esvit/

[4] Swin Pytorch Implementation

[5] Swin Transformer Paper

[6] Dino Paper

[7] Extracting patches from image

[8] Sparse multi head attention