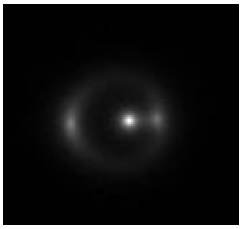
GSoC 2022: Proposal to ML4SCI

Transformers for Dark Matter Morphology with Strong Gravitational Lensing

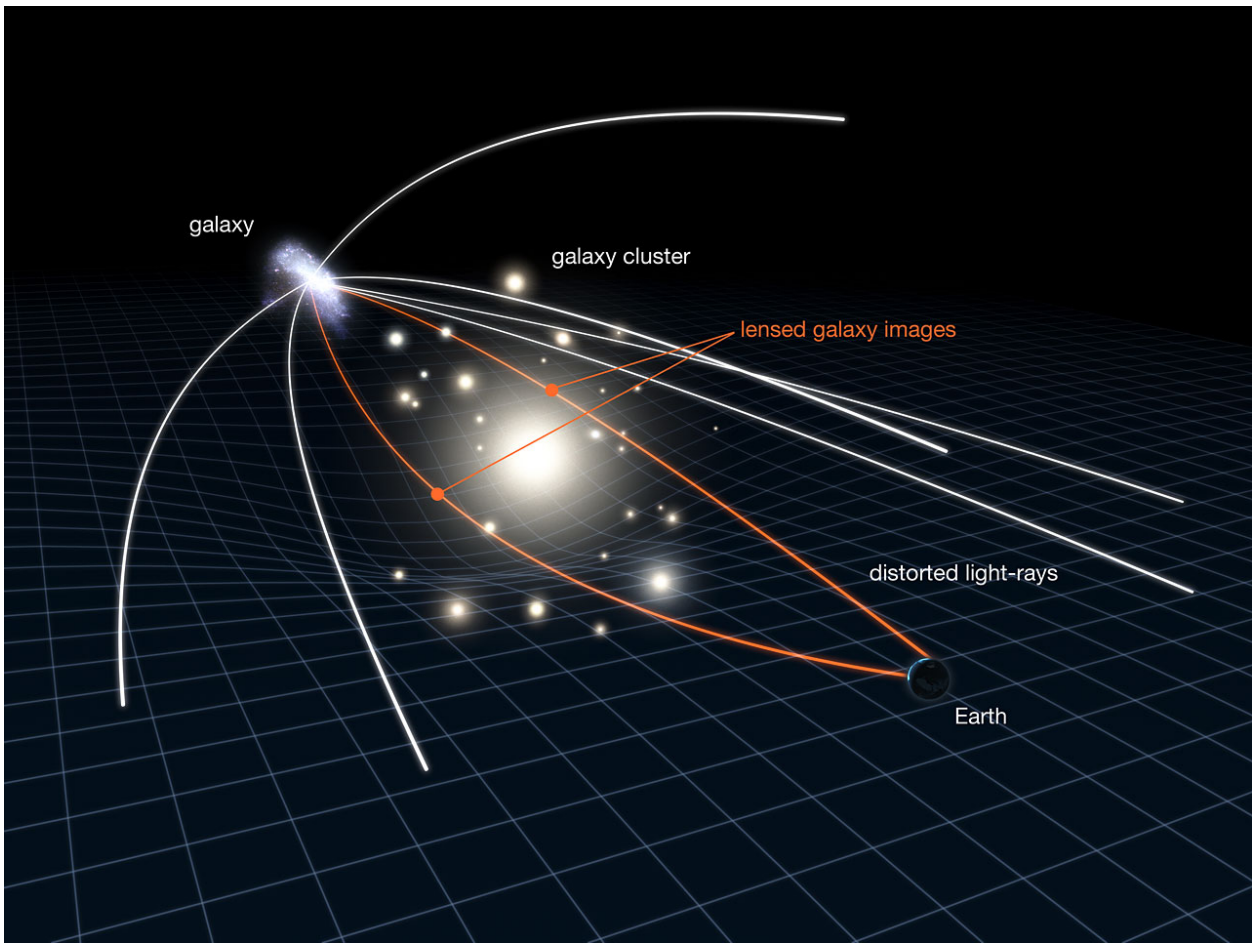
Mentors: Emanuele Usai, Anna Parul, Michael Toomey, Pranath Reddy, Stephon Alexander

DeepLense



INTRODUCTION :

Scientific researches have only been able to ascertain the identity of dark matter from its gravitational fingerprints. A gravitational lens is formed when a huge amount of matter or a cluster of galaxies create a gravitational field that distorts the light from a quasar behind it in the same line of sight. Strong gravitational lensing has delivered promising results in detecting the existence of substructures in dark matter halos. Probing these substructures can help us understand the identity of dark matter and further our understanding of the universe. Deep learning methods have been performed remarkably well in cosmology and also have the ability to distinguish among these substructures representative of different dark matter models. This project aims to implement a vision transformer to probe these galaxy-galaxy strong lensing simulations and also integrate an ergonomic high-level python framework with the DeepLense pipeline for future work.



Gravitational Lensing in action image credits: NASA, ESA & L. Calçada

SCOPE OF THIS PROJECT:

Transformers are semi-supervised learning algorithms that were introduced in 2017 and were limited in the deep learning world to Natural Language Processing(NLP) tasks. Meanwhile the state-of-the-art on image processing tasks was using Convolutional Neural Networks(CNN) until in 2020 when the paper [4] proposed using an iteration of transformers for computer vision problems, which we know today as a Vision Transformer (ViT). By the end of 2021 a pure transformer model would outperform CNNs both on accuracy and efficiency on computer vision tasks.

ViT achieves remarkable results compared to convolutional neural networks (CNN) while obtaining fewer computational resources for pre-training. These transformers try to capture the relationships between different portions of an image by breaking it down to a series of patches similar in some sense to how original transformers measured the relationships between pairs of strings, termed as **attention**. Soon there were several variants of transformers being used at scale on all sorts of computer vision tasks like object detection, image segmentation, autonomous driving, etc.

CNNs have been the main models for image classification since deep learning took off in 2012, but CNNs typically require hundreds of millions of images for training to achieve the SOTA results. Vision transformer models require a lot less data and computing resources for training to compete with the leading CNNs in performing image classification. Accounting the limited data available for strong lensing images and most of the research being powered by simulations, I propose exploring vision transformers to classify the dark matter substructures in strong lensing images.

PROJECT GOALS :

In continuation to the work that has been done on the DeepLense pipeline in using several supervised and unsupervised learning methods for particle dark matter searches in galaxy-galaxy strong gravitational lensing, this project aims to develop a Python framework implementing a vision transformer for SOTA on the image processing task at hand. This project will extend the excellent work described in [3].

By the end of the Google Summer of Code, I plan to have completed the following :

1) A vision transformer capable of SOTA results in classifying strong lensing images into three categories:
    a. images with no substructure,
    b. images with spherical substructure
    c. images with vortex substructure.
2) A python based command line tool to provide a high level interface to generate inferences on new data.
3) A blog post and PPT presentation to summarize the project.

DATASET:

The dataset we will be using is described in [3]. It consists of strong lensing images of different models of DM with disparate forms of substructures which have been categorized into 3 types:

1. **No** substructure.
2. **Spherical** substructure of non-interacting CDM (WIMP).
3. **Vortex** substructure of DM superfluids and condensates.

The images for this dataset were generated by feeding parameters from the paper into the **PyAutoLens** package**.**

EVALUATION METRIC:

We will use the AUC (Area under the Receiver Operating Characteristic curve) as the evaluation metric.

IMPLEMENTATION DETAILS :

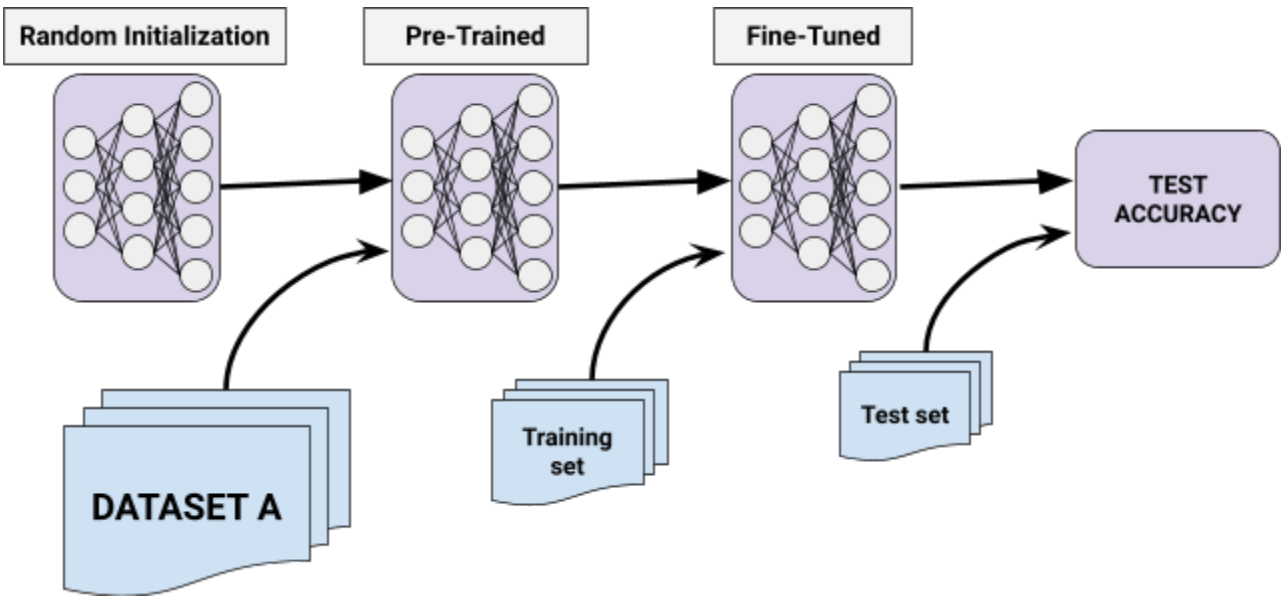
I will be implementing a transformer as described in [4] using the PyTorch framework. The neural network will be optimized using the ADAM [8] algorithm with the learning rate being decayed by a factor of 10 every 20th epoch. The model architecture has been added for reference. To organize the model code, I will be using the PyTorch Lightning framework. I will use matplotlib for visualizations, sklearn for ROC AUC, and numpy and einops for basic tensor operations. Model implementation can be broken down into three main subtasks:

1) **Tokenizing input:** ViT splits an image raster as a series of patches and offers predictions for class labels on the image. To achieve this the strong lensing images will need to be split into patches (left to right and up to down), flattened into 1-Dimensional vectors and then using linear mapping, create an initial trainable position embedding (in the form of a tensor) for each of the patches (see Model Architecture section). In order to add a spatial representation of each patch within the sequence, this embedding learns positional information for each of the patches. To capture the meaning of the image as a whole, we prepend this sequence with a CLASS([CLS]) token which represents an aggregate of the representations of the patches and will be used for classification later.

2) **Transformer Encoder model:** The main idea of a ViT is to use self-attention layers instead of convolutional ones. The ViT model achieves this by using the multi-head self attention in computer vision without requiring the image-specific biases. I will use a Pre-Layer Normalized version of the Transformer blocks to process the tokenized images. Applying Layer Normalization at the first layer in the residual blocks allows for a better gradient flow and also removes the necessity of a warm-up stage. The network will learn

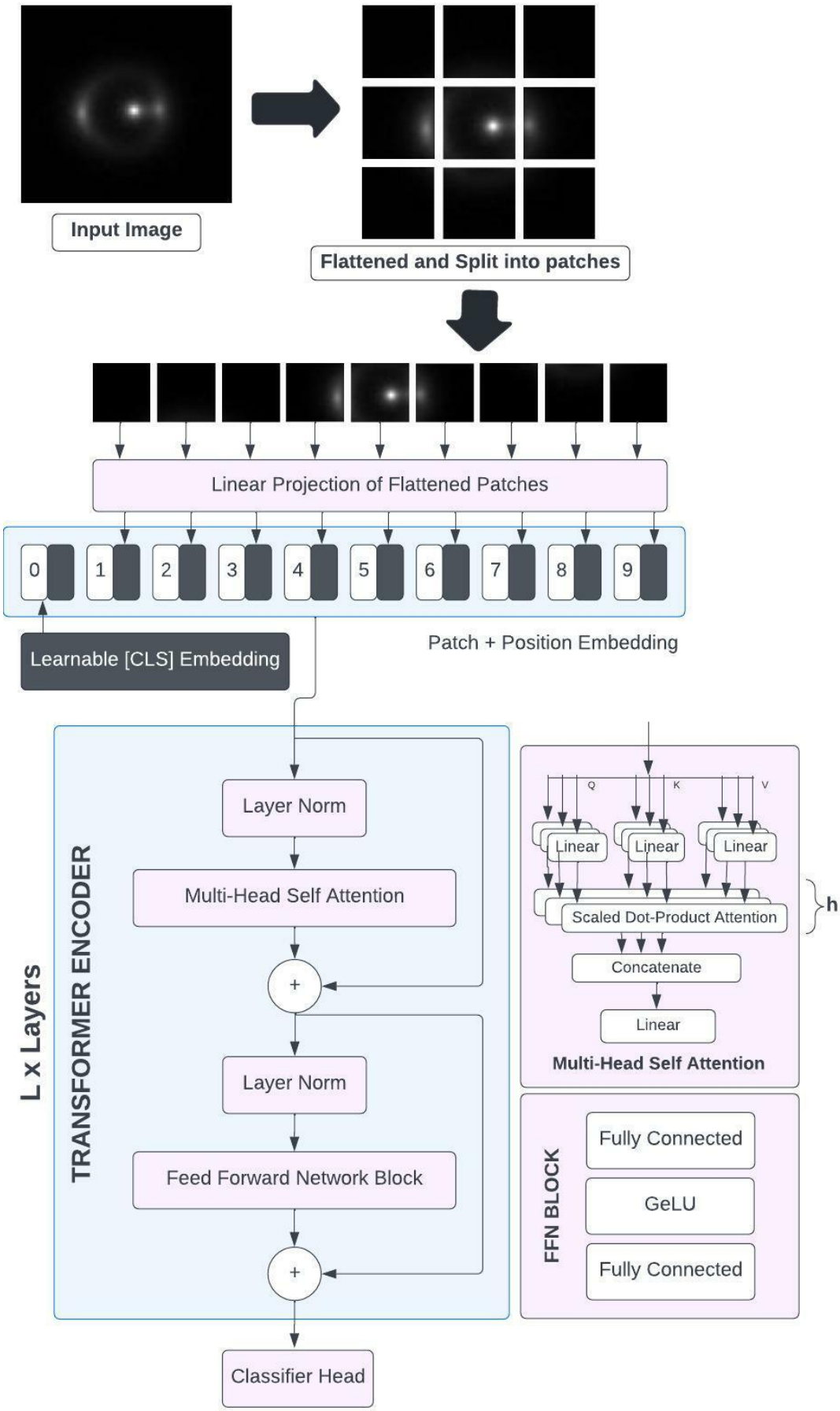
about more abstract features from the embedded patches using a stack of “L” transformer encoders.

3) **Optimization using Tokens:** Structurally, the simplest ViTs lack locality inductive bias due to their poor tokenizing strategy. This means training a model from scratch wouldn’t generate substantial results in the case of small to medium datasets unless the model is pre-trained on huge datasets like JFT-300M. I expect this to be a major benchmarking hiccup later on since the lensing data available to us is very sparse considering ViT standards. Thankfully, ever since the adoption of ViT in the industry, a lot of research has been done to improve its computation benchmarking. [6] [7] among many others propose efficient and sophisticated strategies to overcome the lack of inductive bias issue and I will be implementing one of these methods.
The standard principle for ViT model training is to first train the model on a huge dataset. This step is known as pre-training the model. Then we use our target dataset to generalize the model to our needs. This step is known as fine-tuning the model. We will use this fine-tuned model to evaluate the performance on the test set. We can always skip the pre-training and use already pre-trained model weights available on the internet.



Pre-Training and Fine-Tuning a Machine Learning model

VANILLA MODEL ARCHITECTURE:



PROPOSED SOLUTIONS:

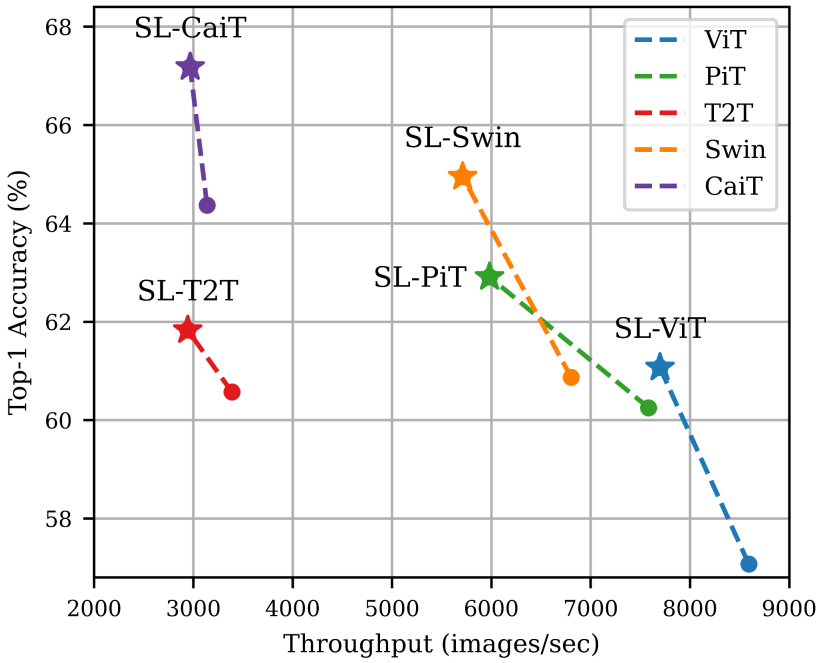
The proposed architecture for an optimized solution would be built on top of the vanilla ViT architecture. Thanks to the evaluation task, almost everything that we will need for the vanilla ViT are housed in my GitHub repository for this task just waiting to be assembled with the add-on modules from the Swin or transformers with SPT LSA. This would help me finish the project relatively earlier. Below I will first try to explain the pretrained solution at hand and then we will continue to discuss the additional components I will be using to optimize the transformer.

Pre-trained solution:

Data-efficient Image Transformers (DeiT) were first proposed in the paper [12]. DeiT is a ViT model pre-trained on ImageNet for image classification. We can make use of the Facebook DeiT which is further described in repo and notebook. These resources have pointers that will be useful for a successful implementation.

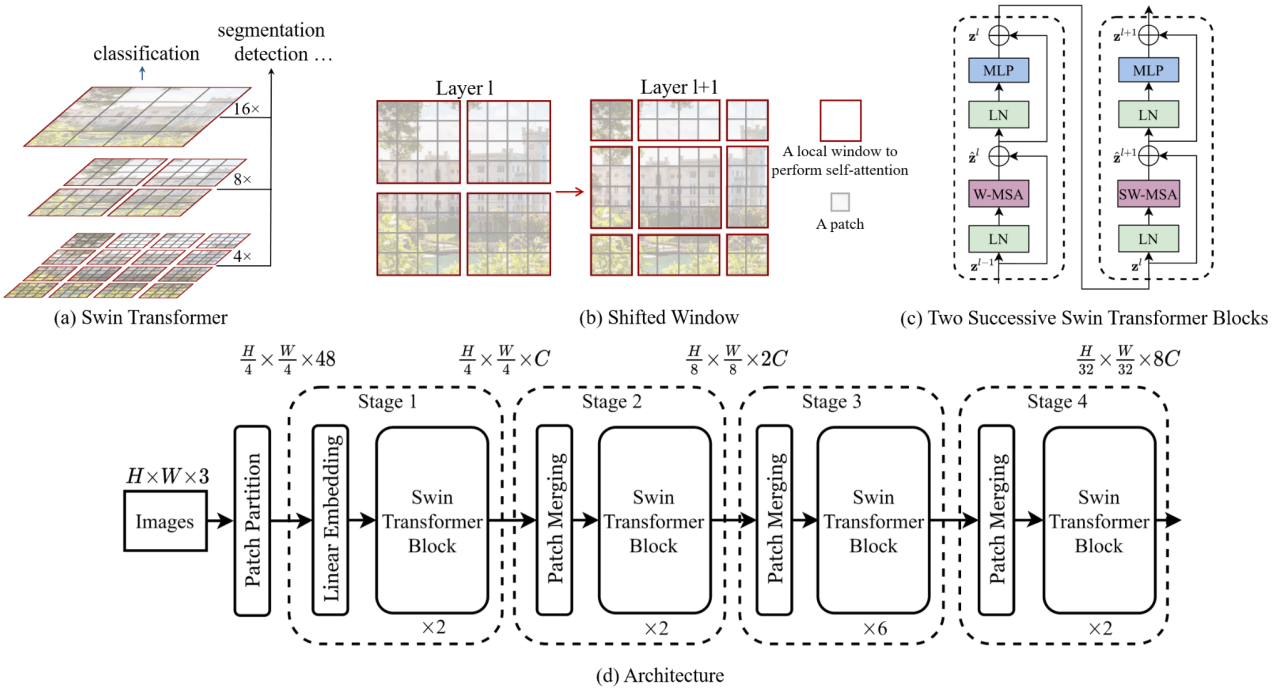
Without Pre-training:

While there are several solutions to fixing the locality inductive bias in ViT, the two most effective ones are discussed below. Swin transformers have excellent results on dense prediction tasks such as object detection and semantic segmentation, and it will be interesting to see their performance on an image classification task. Transformers with SPT LSA are the most effective solution to the lack of locality inductive bias and have benchmarked performance improvements upto 4%.



Accuracy-Throughput Graph on Tiny-ImageNet

Swin Transformer:



(a) Swin Transformer (b) Shifted Window (c) Two Successive Swin Transformer Blocks

(d) Architecture

The Swin Transformer builds hierarchical feature maps similar to Convolutional Neural Networks (a). This model uses patches as in the vanilla ViT model. However, instead of using one size as in ViT (16 by 16px), the Swin Transformer first starts with small patches in the first Transformer layer. As we go deeper into the network, the model merges these layers into bigger ones. It takes an image and splits it into n pixel by n pixel patches (usually 4). Each patch may be an image with c channels. Thus, a patch has a total of n*n*c feature dimensionality. It is then linearly transformed into a desired dimensionality called C.

As we can see in (d), the major architectural difference in the vanilla ViT and Swin is the replacement of the standard multi-head self-attention (MSA) module for a module based on shifted windows in a transformer block, with the remaining layers remaining unchanged. Hence, a shifted transformer block comprises a shifted window-based MSA module, a two-layer MLP, and GELU nonlinearity. Each MSA and MLP module is preceded by a LayerNorm (LN) layer, and a residual connection follows each module (c).

ViT with SPT LSA:

The proposed Shifted Patch Tokenization (SPT) and Locality Self-Attention (LSA) effectively solve the lack of locality inductive bias and enable vision transformers to learn from scratch even on small-size datasets. Moreover, SPT and LSA are generic and effective add-on modules that are easily applicable to various ViTs.

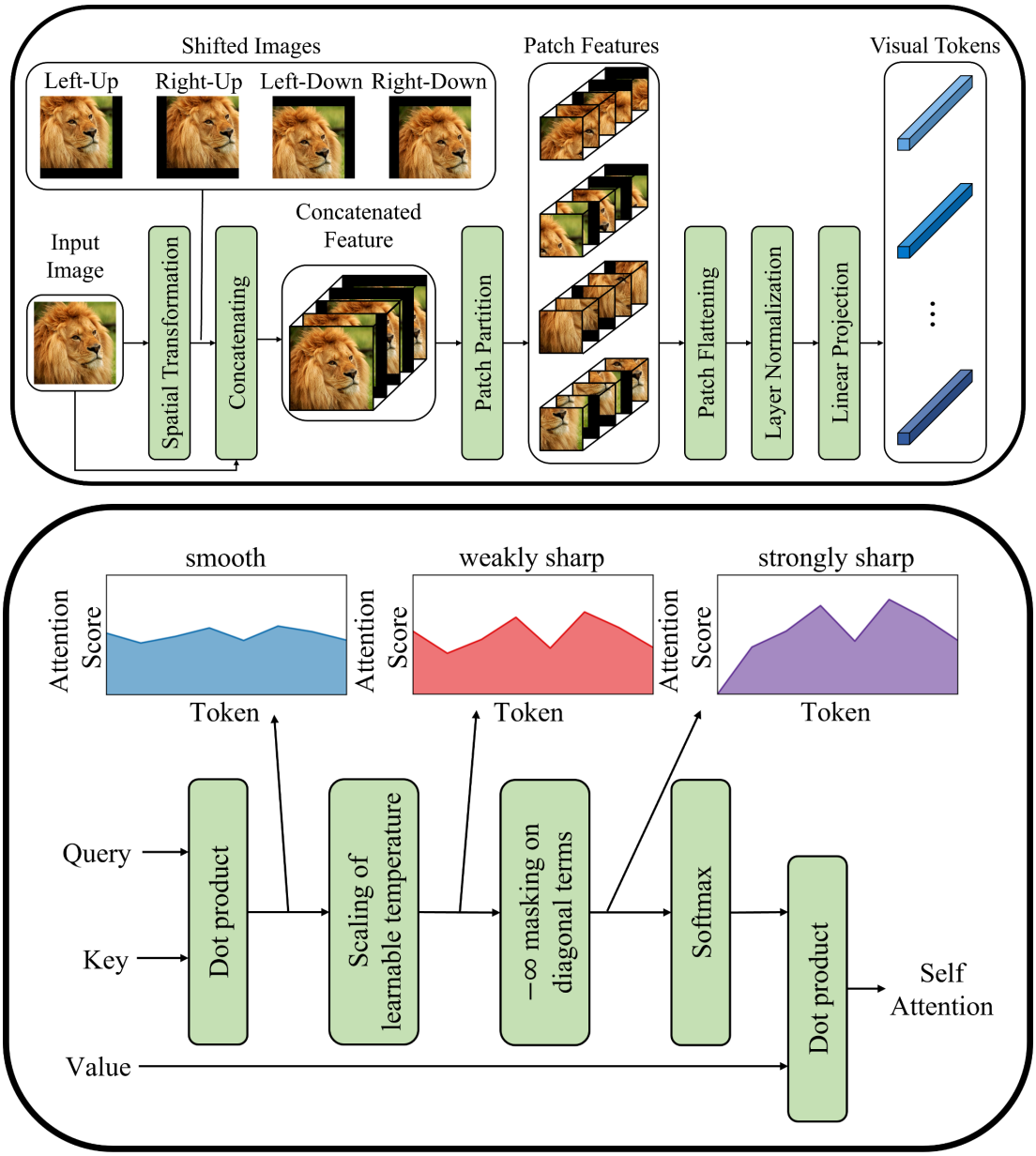


Image Credits: Vision Transformer for Small-Size Datasets [6]

TIMELINE :

To conform the project to GSoC timeline I have broken the monolithic task into subtasks and have listed them in an approximation according to major features that need to be implemented. There might be unforeseen issues which may or may not take more time than allotted, but generally I would wish to finish everything well before time and reserve the last two weeks to test the framework for the best performance and adding more features.

Community Bonding Period (May 20 - June 12):

- ➢ Interacting with the mentors and students in the community.
- ➢ Collecting feedback regarding my proposal and scope of implementation.
- ➢ Subsume new insights and explore other transformer architectures and prepare a final goto plan.
- ➢ Reading through the recommended research paper reading list.

Deliverables:

- ➢ Blog post.

Phase 1 (June 13 - July 25)

- ➢ The main focus in my phase 1 timeline is to have a well performing model ready by the end of the first month.

Week 0-3:

- ➢ Assimilate any new ideas and the new changes (if any) and start preparing methods to load data into a custom DataSet and adding data preprocessing pipelines (if needed).
- ➢ Then I would write methods to implement data logging features.
- ➢ Since we will be working with image data, developing a robust data transformation pipeline is very critical.
- ➢ Loading data as a custom DataLoader.
- ➢ Discuss with the mentors about interesting visualization strategies for strong lensing images (and images for the scientific community in general).
- ➢ Begin working on the model.

Deliverables:

- ➔ All data is expected to load into a DataLoader object with the proper folder class structure.
- ➔ A seamless control over image data using the DataLoader object and image array rasters.
- ➔ Ability to generate elucid illustrations of the data.

Week 4-6:

- ➢ Finish the model and train the first iteration of the finished model on the training data.
- ➢ Improvements based on feedback received.
- ➢ Work on the CLI framework and implement the various arguments for the tool.

Deliverables:

- ➔ Initial ViT Model and first set of model evaluation results.
- ➔ Ability to interact with the model from the CLI.

Phase 1 Evaluations

- ➢ RE: Expected to finish with an implementation of ViT and discussions regarding eager model evaluation results and optimizations to follow.

Deliverables:

- ➔ Blog Post summarizing Phase 1 progress.
- ➔ Complete Phase 1 report and documentation.

Phase 2 (July 25 – September 5)

- ➢ The focus for phase 2 will be dedicated to fine tuning the model.
- ➢ As discussed in the Project Implementation section, I expect the need to develop either a Swin Transformer or a Shifted Patch Tokenization (SPT) and Locality Self-Attention (LSA) enabled transformer. These strategies are very modular and can be incorporated into the already existing transformer code relatively easily.

Week 6-9:

- ➢ Implement the methods for multiclass ROC-AUC and Confusion Matrix metrics.
- ➢ Implement an optimized version of ViT using modular components.
- ➢ Integrate the ROC-AUC curve generation and Confusion Matrix CLI handlers.

Deliverables:

- ➔ ROC AUC metric for monitoring the model performance and rich integration with the toolkit.

Week 9-12:

- ➢ Finalize the example notebook and other important documentation files (README.md, INSTALL.md, etc).
- ➢ Complete any pending work, code cleanup and fix any final bugs.
- ➢ Increase the habitability of code.
- ➢ Improvements based on the feedback from mentors and community members.

Deliverables:

- ➔ Blog post.

Code Submission and Second Evaluation

Deliverables:

- ❖ A robust code pipeline using ViT which is able to determine the dark matter substructures in simulated strong lensing images.
- ❖ A set of serialized model parameters for future inference.

Results Announced September 20

Post GSoC period:

After the GSoC program, I would like to implement a regression algorithm using Vision Transformers to learn the mapping between the lensing images and the lensing dark matter halo mass. This problem was a part of the ML4SCI hackathon and also the evaluation tasks and I would very much like to benchmark the performance of a vision transformer for the same.

ABOUT ME :

Full Name: Swapnil Tripathi

Timezone: UTC +0530 Kolkata/India

Mobile: +919140589530

Email address: swapnil06.st@gmail.com

Freenode IRC nick: swaptr

Location: Lucknow, India

Education:

Institution: Kalinga Institute of Industrial Technology, Bhubaneshwar, India

Degree: Bachelor of Technology in Electronics and Computer Science

Current Year: 3

PARTICIPATION :

Time:

This will be a **MEDIUM** project and I will be requiring **175** hours to finish this project.

Availability:

I am available online from 5:30am to 6:30pm UTC.

Communication:

I realize the importance of communication so every week I will report to mentors the work done and plans for the next week. I will be available to communicate on a daily basis both to the mentors and to other students on email and IRC.

Workflow:

I will be working on a separate branch on git and uploading code to the forked repository every two days.

Commitments:

I have no other commitments this summer, I will be able to give 30-35 hours per week and I am ready to commit more in order to finish the goals of the project.

I am not submitting any other proposals.

REFERENCES:

1. Alexander, Stephon H. S. et al. “Domain Adaptation for Simulation-Based Dark Matter Searches Using Strong Gravitational Lensing.” (2021).
2. Alexander, Stephon H. S. et al. “Decoding Dark Matter Substructure without Supervision.” *arXiv: Cosmology and Nongalactic Astrophysics* (2020): 18.
3. Alexander, Stephon H. S. et al. "Deep Learning the Morphology of Dark Matter Substructure."(2020): 15.
4. Dosovitskiy, Alexey et al. “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale.” *ArXiv* abs/2010.11929 (2021): 22.
5. Vaswani, Ashish et al. “Attention is All you Need.” *ArXiv* abs/1706.03762 (2017).
6. Lee, Seung Hoon, Seunghyun Lee, and Byung Cheol Song. "Vision Transformer for Small-Size Datasets." arXiv preprint arXiv:2112.13492 (2021)
7. Liu, Ze et al. “Swin Transformer: Hierarchical Vision Transformer using Shifted Windows.” *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (2021): 9992-10002
8. Kingma, Diederik P. and Jimmy Ba. “Adam: A Method for Stochastic Optimization.” *CoRR* abs/1412.6980 (2015): 15.
9. Fayyaz, Mohsen et al. “Adaptive Inverse Transform Sampling For Efficient Vision Transformers.” (2021).
10. Wang, Wenxiao et al. “CrossFormer: A Versatile Vision Transformer Hinging on Cross-scale Attention.” (2021).
11. Chu, Xiangxiang et al. “Twins: Revisiting the Design of Spatial Attention in Vision Transformers.” (2021).
12. Touvron, Hugo et al. “Training data-efficient image transformers & distillation through attention.” *ICML* (2021).