GSoC 2023 Project Proposal

Organization : ML4SCI

Gravitational Lens finding for Dark Matter substructures

AUTHOR

SAINITHIN ARTHAM

Contents

1 Introduction and Student Information

- Name: SAINITHIN ARTHAM
- Email: sainithin artham , sai.artham.19cse@bmu.edu.in
- GitHub: Sai Nithin-bit
- Contact no: +91-9494017463 / +91-8074098298
- University: BML Munjal University (Delhi), University of Warwick (Uk)

  Major: Computer Science

  Minors: Computational Mathematics

  Summer : Game Theory and competitive strategy

2 Pre Test Results

Common Test

This project implements Semantic Clustering by Adopting Nearest Neighbors (SCAN) algorithm (Van Gansbeke et al., 2020). The algorithm consists of two phases:

1. Self-supervised visual representation learning of images, in which we use the simCLR technique.
   simCLR (Simple Contrastive Learning of Representations) is a method for self-supervised learning of visual representations. It works by maximizing the similarity between augmented views of the same image and minimizing the similarity between views of different images. The idea is to encourage the model to learn to extract meaningful features that are invariant to various image transformations such as rotation, color distortion, and cropping.

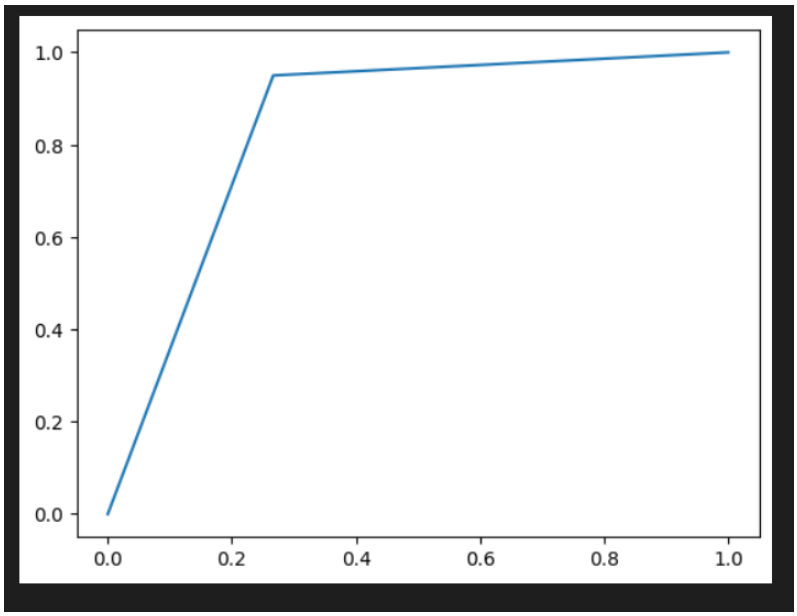
   The learned representations can be used for various downstream tasks such as object detection, image classification, and segmentation. simCLR has been shown to achieve state-of-the-art performance on several benchmark datasets, even when compared to supervised methods.

2. Clustering of the learned visual representation vectors to maximize the agreement between the cluster assignments of neighboring vectors.

   In the context of learned visual representation vectors, clustering can be used to identify patterns or groupings in the representations. When clustering learned visual representations, the goal is to maximize the agreement between the cluster assignments of neighboring vectors, we can ensure that the resulting clusters are meaningful and useful for the intended application.

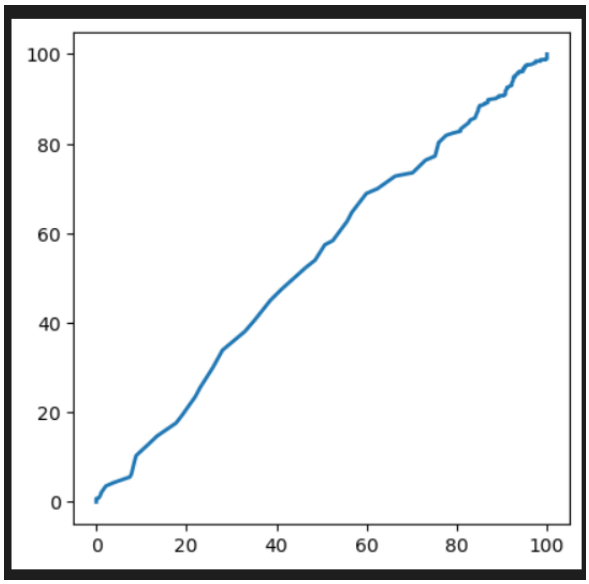
   For example, if we want to classify new images based on their visual similarity to a set of reference images, we can assign the new images to the cluster with the highest similarity to the reference images.

Roc Metric plot of TPR vs FPR on common test. For more details please refer to notebook names common_test



Specific Test - 2

This Project implements the Vision Transformer (ViT) model by Alexey Dosovitskiy et al. for image classification. The ViT model applies the Transformer architecture with self-attention to sequences of image patches, without using convolution layers. Images in the Lense_data contain a lot of noise in which the traditional CNN considers this noise as part of the image and tries to extract features. It may be misleading for our model and needs a lot of training data for mitigating this issue.CNNs process input images in a local and hierarchical way, which means that they have a limited view of the context surrounding an object or feature. This can make it challenging for CNNs to recognize objects or features that are partially occluded or located in cluttered environments. Whereas transformers come in handy for capturing long range dependencies of the image and provide better results even when features are cluttered over the image.The reasons for choosing transformers is , this area is not explored for this task , i have chosen basic version of transformers for this task , however there are more advanced heavily pretrained models like Swin Transformers [5]. The results appear to further explore this area. Keeping in mind of large data sets in future, transformers can be easily parallelized across multiple GPU's then CNN's. Given below is ROC metric plot of TPR vs FPR on specific test , for details please refer to notebook named specific_test



3 Abstract

Strong gravitational lensing systems are unique systems in which a background galaxy and a foreground galaxy or cluster of galaxies are sufficiently well-aligned so that the gravitational field of the foreground system lenses the background galaxies. Whilst these lensing systems hold a rich source of information of the gravitational field distribution of the foreground system and can be used to map dark matter distribution within the cluster, they are rare to come by. Traditional methods for detecting these strongly lensed systems were based on visual inspection and this proposal aims to address the automation of this detection.

As we enter the era of large-scale imaging surveys with the telescopes such as LSST and SKA, it is envisaged that the number of known strong gravitational lensing systems will increase dramatically. However, these events are still very rare and require the efficient processing of millions of images. In order to tackle this image processing problem, I propose Deep Learning techniques and apply them to the Gravitational Lens Finding Challenge.

4 Technical Details

a. Project Overview

Denoising techniques can be applied to images of strong and weak gravitational lenses to improve the accuracy of their classification. Strong gravitational lenses produce highly distorted images of background sources, while weak gravitational lenses produce subtle distortions that are more difficult to detect. Accurate classification of these lenses is essential for studying the properties of dark matter and the structure of the universe. In many cases, images of gravitational lenses are corrupted by noise, which can make it difficult to accurately classify them. Denoising techniques can be used to remove this noise and improve the signal-to-noise ratio of the images, making it easier to detect and classify the distortions caused by gravitational lenses. One approach is to use deep learning-based denoising algorithms, which can learn to remove noise from images while preserving the important features that are relevant for lens classification. I propose to use Unet[3] architecture to accomplish this task. [3] has done extensive analysis on before and after denoising improvements on astronomy images, i am unable to mention here due to space constraints, please feel free to check them out

A new strong gravitational lens finder is important for many downstream tasks. The goal was to reduce the need for human inspection and increase efficiency in the identification and classification of strong gravitational lenses.
In this project I propose to implement 3 deep learning based models.

1. Pretrained Swin Transformer
2. LASTRO EPFL Model
3. CMU Deeplense

3 TECHNICAL DETAILS

I am proposing to add pre trained Swin Transformer model to the repository for strong lens classification.It was introduced in the paper "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows" by Microsoft Research. Swin Transformer achieves state-of-the-art performance on several image classification benchmarks while being computationally efficient.

LASTRO EPFL and CMU Deeplense, The reason for proposing these 2 CNN models is, they are specifically developed for gravitational strong lens classification rather than pre-trained on general data and fine tuning.[1] model has achieved accuracy over 95 % in Gravitational lens finding challenge.

[1] has mentioned many variants of CNN's , however baseline architecture is out-performing over all others, which I am proposing to implement.

Space	Test AUC	$\text{Recall}_{0\text{FP}}$	$\text{Recall}_{1\text{FP}}$
baseline	0.9322 ± 0.0016	0.01 ± 0.02	0.04 ± 0.04
committee b.	0.9326	0.01	0.01
views	0.9324 ± 0.0013	0.26 ± 0.06	0.28 ± 0.07
committee v.	0.9343	0.30	0.32
residual	0.9322 ± 0.0006	0.23 ± 0.04	0.29 ± 0.03
committee r.	0.9346	0.29	0.30
invariant	0.9332 ± 0.0006	0.27 ± 0.04	0.28 ± 0.05
committee i.	0.9399	0.32	0.33
Ground	Test AUC	$\text{Recall}_{0\text{FP}}$	$\text{Recall}_{1\text{FP}}$
baseline	0.9761 ± 0.0011	0.44 ± 0.13	0.49 ± 0.08
committee b.	0.9773	0.50	0.55
views	0.9746 ± 0.0011	0.35 ± 0.19	0.43 ± 0.17
committee v.	0.9759	0.35	0.39
residual	0.9775 ± 0.0006	0.44 ± 0.06	0.46 ± 0.07
committee r.	0.9795	0.50	0.55
invariant	0.9774 ± 0.002	0.39 ± 0.11	0.45 ± 0.05
committee i.	0.9813	0.49	0.49

Fig 1

CNNs variants and reported accuracy presented in [1]

Confusion Matrix for Baseline Architecture[1]

Space-based	Classified as non-lens	Classified as lens
Non-lens	59742	40
lens	20957	19264
Ground-based	**Classified as non-lens**	**Classified as lens**
Non-lens	50042	17
lens	21754	28194

b. Dataset Description

In this project I propose to use Bologna lens factory data, The Bologna Lens Factory is an international collaboration of scientists who aim to study strong gravitational lenses in astronomical surveys. The collaboration organized a strong-lensing challenge that involved analyzing data from ground-based and space-based surveys to identify and classify strong gravitational lenses.

The data used in the Bologna Lens Factory challenge consisted of images of the sky taken using different telescopes and filters. The ground-based data set was obtained from the Canada-France-Hawaii Telescope (CFHT) and was composed of 6,000 images in four different bands: u, g, r, and i. The space-based data set was obtained from the Hubble Space Telescope (HST) and consisted of 170 images in the F606W and F814W filters.

Each image in the data sets contained potential strong lens candidates, which were identified based on their elongated shape or the presence of multiple images of a single object. The data set has two types of sets: ground based and space based.Training set of 20,000, 101px x 101px images are present for space based and 20,000 x 4 images are in ground based training set in four bands. For each image in the training sets a classification as a lens or not lens is provided in a ASCII log file

c. Methodology [Framework of Implementation]

Data preparation is common for both denosing and classification, so it is done once in the whole procces.

Data preparation

1. Common Steps

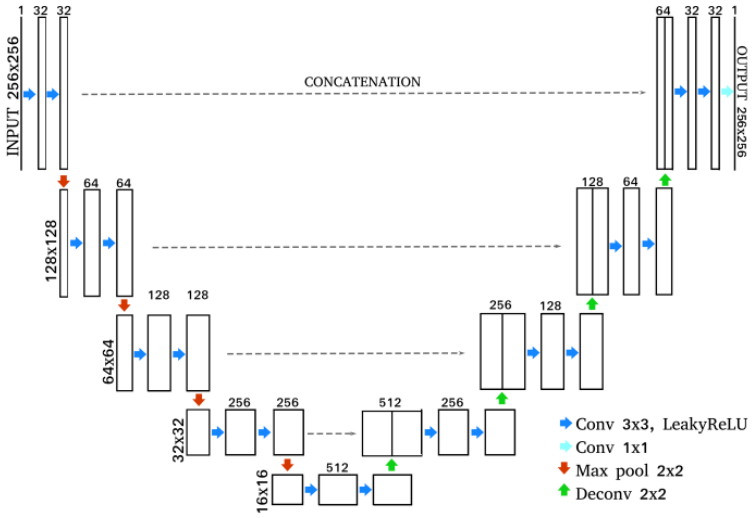
   Normalization , resizing, splitting, balancing data.

2. Data Augmentation

   CNNs are, by design, already invariant to translation, but not to rotation, scaling, and flipping. So this step is crucial for a robust model.The benefit of increasing the training set size is to reduce overfitting.The image augmentation component utilizes these 9 different transformation

   - A Vertical or Horizontal flipping of the image.
   - A 90°, 180°or 270°Rotation of the image.
   - A Translation of [-10%, 10%] of the image along the X and/or Y axes
   - A Scaling of [0.75, 1] of the image along the X and/or Y axes
   - A Shearing of [-20%, 20%] of the image along the X and/or Y axes

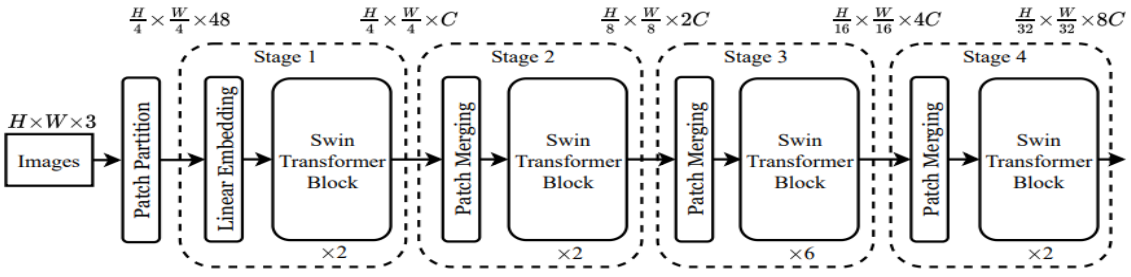
Denoising Model [Denosing Framework]



Denoising data

Adding an additional step of denoising with a proposed denoising model can potentially improve the accuracy of classification of strong and weak gravitational lenses. To apply this approach, we first obtain the noisy image and then apply the proposed denoising model to remove the noise. The denoised images are then used for classification.

Swin Transformer [Swintransformer Framework]
Model architecture.



CMU Deeplens model

```
x = input

x = keras.layers.Conv2D(32, (7, 7), padding='same', activation='elu')(x)
x = keras.layers.BatchNormalization()(x)

x = resnet_block(x, 16, 32, False)
x = resnet_block(x, 16, 32, False)
x = resnet_block(x, 16, 32, False)

x = resnet_block(x, 32, 64, True)
x = resnet_block(x, 32, 64, False)
x = resnet_block(x, 32, 64, False)

x = resnet_block(x, 64, 128, True)
x = resnet_block(x, 64, 128, False)
x = resnet_block(x, 64, 128, False)

x = resnet_block(x, 128, 256, True)
x = resnet_block(x, 128, 256, False)
x = resnet_block(x, 128, 256, False)

x = resnet_block(x, 256, 512, True)
x = resnet_block(x, 256, 512, False)
x = resnet_block(x, 256, 512, False)

x = keras.layers.AvgPool2D()(x)

x = keras.layers.Flatten()(x)

output = keras.layers.Dense(1, activation='sigmoid')(x)

model = keras.models.Model(input, output)
```

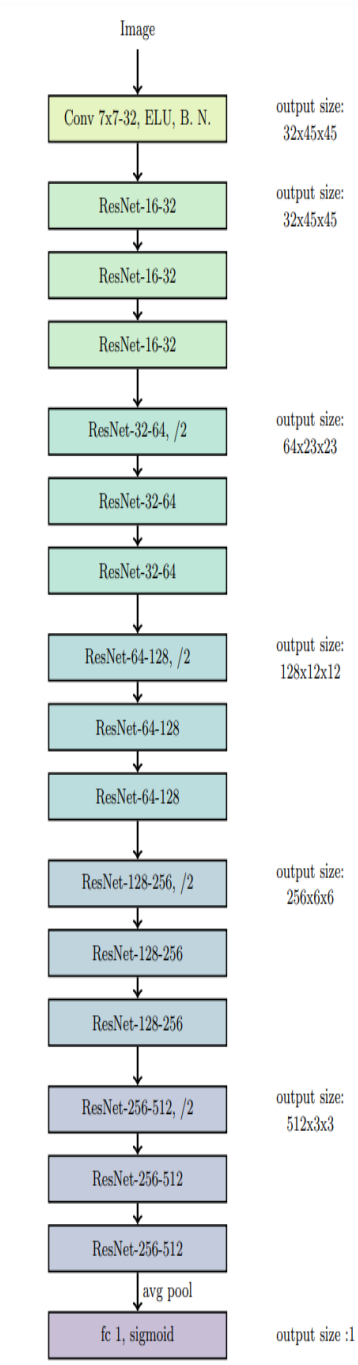


Fig taken from original paper [4]

LASTRO EPFL Model

```
model.add(keras.layers.Conv2D(16, (4, 4), padding='valid', strides=1, activation='relu',
                              input_shape=(image_size, image_size, no_of_channels)))
model.add(keras.layers.Conv2D(16, (3, 3), padding='valid', strides=1, activation='relu'))
model.add(keras.layers.MaxPool2D(pool_size=(2, 2)))
model.add(keras.layers.BatchNormalization())

model.add(keras.layers.Conv2D(32, (3, 3), padding='valid', strides=1, activation='relu'))
model.add(keras.layers.Conv2D(32, (3, 3), padding='valid', strides=1, activation='relu'))
model.add(keras.layers.MaxPool2D(pool_size=(2, 2)))
model.add(keras.layers.BatchNormalization())

model.add(keras.layers.Conv2D(64, (3, 3), padding='valid', strides=1, activation='relu'))
model.add(keras.layers.Conv2D(64, (3, 3), padding='valid', strides=1, activation='relu'))
model.add(keras.layers.MaxPool2D(pool_size=(2, 2)))
model.add(keras.layers.BatchNormalization())

model.add(keras.layers.Dropout(0.1))

model.add(keras.layers.Conv2D(128, (3, 3), padding='valid', strides=1, activation='relu'))
model.add(keras.layers.Dropout(0.1))

model.add(keras.layers.Conv2D(128, (3, 3), padding='valid', strides=1, activation='relu'))
model.add(keras.layers.BatchNormalization())
model.add(keras.layers.Dropout(0.1))

model.add(keras.layers.Flatten())

model.add(keras.layers.Dense(1024, activation='relu'))
model.add(keras.layers.Dropout(0.1))
model.add(keras.layers.Dense(1024, activation='relu'))
model.add(keras.layers.Dropout(0.1))
model.add(keras.layers.Dense(1024, activation='relu'))
model.add(keras.layers.BatchNormalization())

model.add(keras.layers.Dense(1, activation='sigmoid'))
```

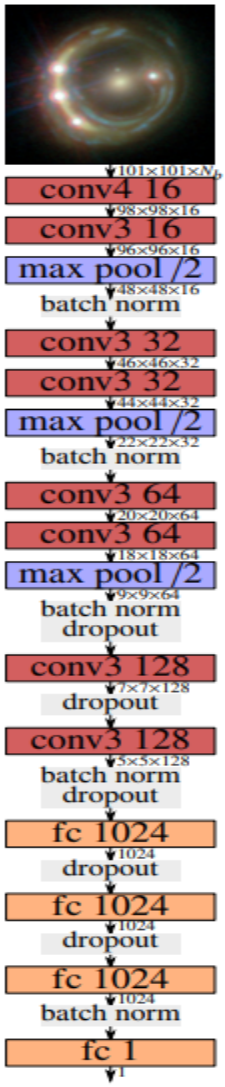


Fig taken from original paper [1]

Training

1. Model Building

   Defining Architecture and standard operations.

2. Metrics passing

   Receiver Operating Characteristic (ROC) is a widely used metric for evaluating the performance of binary classification models. ROC curves and AUC-ROC are useful metrics for evaluating the performance of binary classification models in situations where the cost of false positives and false negatives is different.

3. Other Settings
   - Loss - Binary cross entropy loss function
   - optimizer - Adam optimizer
   - learning rate - 0.001
   - weight decay - 0.0001
   - Batch_size - 512
4. Model Saving

   Saving the trained model weights in a pickle file for reusing.

Testing

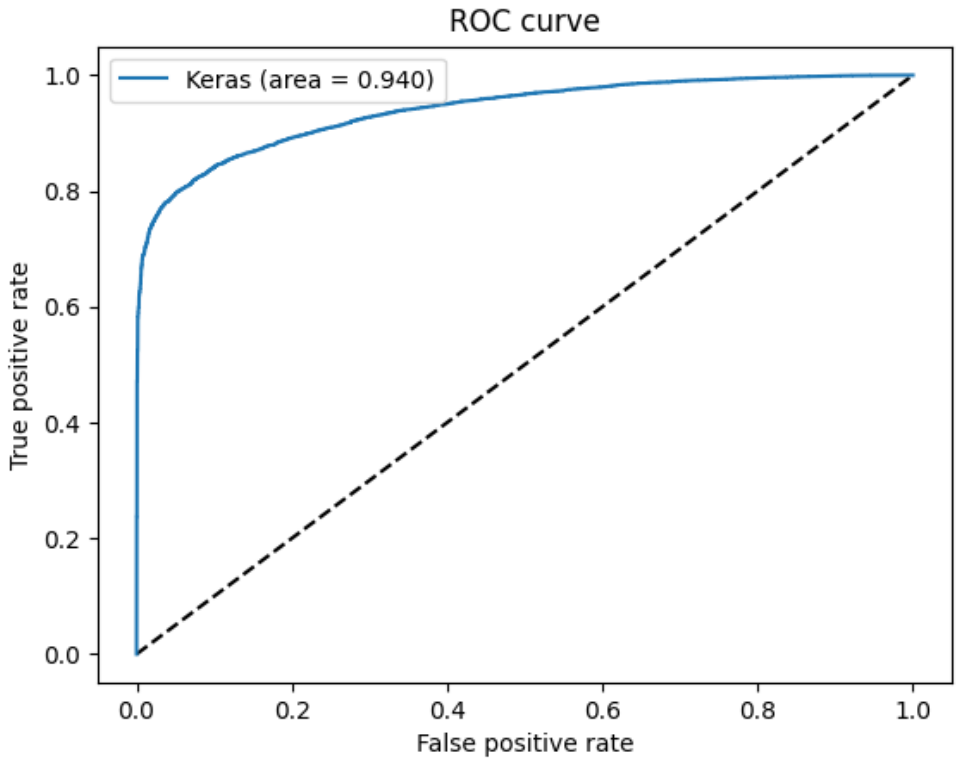
Finally, the trained CNN models can be tested on the testing set to evaluate its real-world performance. The model's performance can be measured using the same metrics as used in the validation step.

This involves passing the preprocessed images through the model and obtaining the predicted classes or probability scores. These probability scores are used for evaluation and can be done using various metrics such as accuracy, precision, recall, F1 score, and

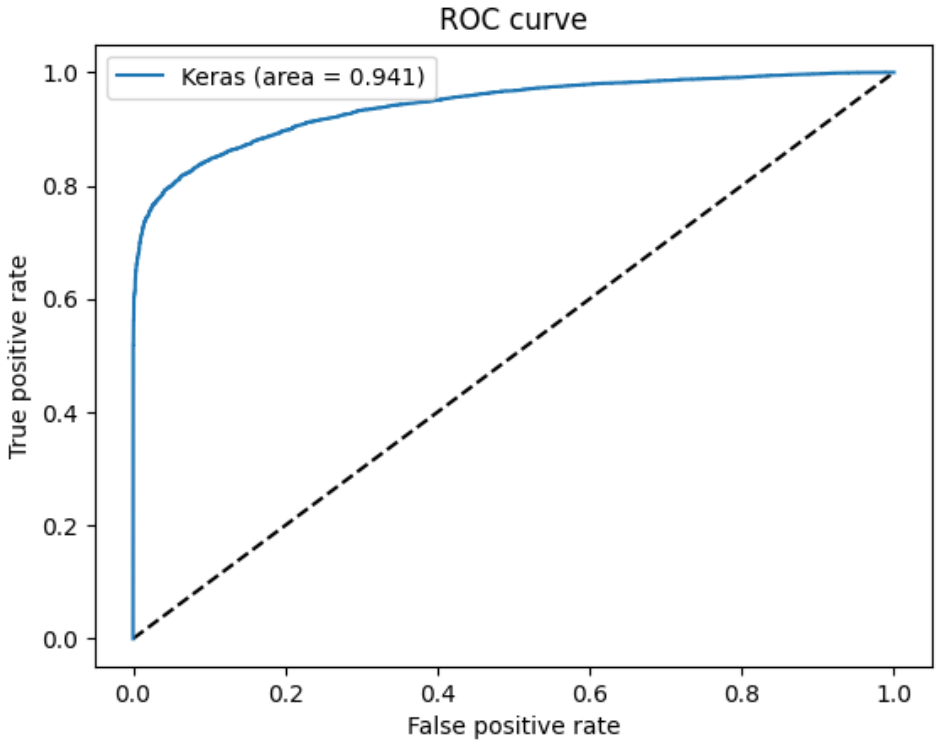
ROC curve. These metrics help to determine how well the model generalizes to unseen data.

Results

Results of specific-test data on LASTRO-EPFL pretrained model. Only test set(1800 images).For more details check my github repo.



Results of specific-test data on CMU-Deeplens pretrained model. Only test set(1800 images).For more details check my github repo.



After GSOC

I have also seen some other potential models for strong lens classification , like

1.WSI_NET [Ref]

2.CAS_Swinburn [Ref]

3.Lens_flow[Ref]

But gsoc has limited time to include them in my proposal, i guess i will add them in separate PR with approval of organisation and discussing it with mentors.The goal is include a wide range of quality models for user flexibility and reducing their time more on how to use and encouraging researchers to think about where to use.(Irrespective of my acceptance to gsoc i will do this)

5 Proposed Deliverables

- Implementing a general denoising framework with main focus on gravitational lensing classification but not limited only to it.
- Implement robust binary classification models (LASTRO EPFL, CMU Deeplens, Swin Transformer)
- Explore domain adaptation methods for proposed models.
- Generate new datasets for deeplens pipeline.
- Report results on various deeplens classification ,regression and anomaly detection pipelines on found strong lens images.

A. Schedule of Deliverables

a. Community Bonding Period: June 1, 2021 - June 7, 2021

During the community bonding period, I will look into the relevant literature useful for the project and brush up on various techniques for dealing with strong lens images. I will also do some research on best practices for implementing deep learning models and making integration into various pipelines seemlesly.

b. Week 1

Work on preparing data for the model. Implement data augmentation and visualization of data . Balancing of data classes , resizing the data. Developing specific classes and functions for data preprocessing.

c. Week 2

Work on developing proposed denoising model . Model architecture, model trainer, data loaders, classes are to be implemented.

d. **Week 3**

   Work on Swin Transformer binary classification model.

e. **Week 4**

   Work on LASTRO EPFL binary classification model[1].

f. **Week 5**

   Complete all previous tasks.This is a buffer week for any unprecedented delays. Publish a blog post. Prepare for Phase 1 Evaluation

---

Phase 1 Evaluation

g. **Week 6**

   Work on Binary classification model CMU Deeplens[4].

h. **Week 7**

   Generate new dataset , report the results of each pipeline on new dataset by providing a detailed analysis of their performance, highlighting any differences in accuracy or performance between the various pipelines.

i. **Week 8**

   Explore ,adapt and assess domain adaptation methods for proposed models. Assess and Verify denoising model's impact.

j. **Week 9**

   Complete documentation for newly built methods, verify results, fix bugs (if any) and write additional unit tests.

k. **Week 10**

   Complete Jupyter Notebook Tutorials for all the proposed methods and modifications. Publish blog posts and prepare for Phase 2 Evaluation.

Phase 2 Evaluation

A buffer of two weeks are kept for any unwanted delays

6 Other Information

a. Why ML4SCI?

I believe that Python is currently the juggernaut in science and engineering and the current leader in ML, AI, Pharmaceuticals, etc. ML4SCI being the organization dedicated to using machine learning in the basic sciences, I believe lens finding is an important tool for astronomers to study the universe, from mapping dark matter to detecting exoplanets and testing theories of gravity. I believe that the broader impact of my work with the additional feature of using AI will greatly help researchers and practitioners to explore a plethora of other phenomena and moreover significantly simplify their work. Lastly, as an advocate for open-source, it would indeed be an absolute honor to contribute these tools for open science and society.

Relevant Background

b. Past Experience

I have a reasonably good background in Deep Learning. I am an advanced Python user with 4 years of experience in working with the design, development, and deployment of Machine learning and Deep Learning models and currently work as an undergraduate researcher at the Complex Systems and Quantum Information Lab at IISER Pune, India. I have previously contributed to many open source organizations, namely:

i. mlpack : Generation Multi objective optimization algorithm (#PR1 Not merged, #PR2)

ii. Ceph : Contributor to documentation (#PR Merged)

iii. Research Assistant : Using Machine Learning (Tree based ML) for peptide analysis(Link)

iv. Paper published - Metro made easy(Title) ,Integrated a path finding algorithm into a metro app, determining the optimal route between two given points, and providing information to the user on which metro stations to cross and where to change to another train.(Certificate)

Currently I am working on my thesis in undergraduate involving transformers and multimodal learning for video captioning.

5.3.1 Knowledge of Libraries and/or Technology stacks

As for the familiarity with technology stacks, I have extensively used Google's Tensorflow, used Pytorch framework for projects related to deep learning. Keras and scikit-learn are my go-to Deep/ Machine learning libraries. Numpy is the library that I have used most in my projects. Since all the above mentioned libraries are written in Python, I am fluent with Python programming in general.

c. Other commitments

My final-year thesis presentation gets over by the end of May and post-that, I will be able to spend 40 hours per week throughout the summer on this project.

7 References

[1] https://arxiv.org/pdf/1705.07132.pdf

[2] https://arxiv.org/abs/2005.12320

[3] https://arxiv.org/pdf/2011.07002.pdf

[4] https://ui.adsabs.harvard.edu/abs/2018MNRAS.473.3895L/abstract

[5] https://arxiv.org/pdf/2103.14030.pdf