

# Tests for Prospective GSoC 2025 Applicants

Below are the tests we will use to evaluate prospective GSoC students for the [DeepLense](#) projects. After completing the first common test, please thoroughly complete the specific second test for your project of interest and (optionally) other second tests if you would like to also be considered for additional DeepLense projects at the same time (this may increase your chances of success, but make sure you don't do it at the expense of the specific project you are interested in)

**Note:** please work in your own github branch (i.e. NO PRs should be made). Send us a link to your code when you are finished and we will evaluate it.

Please **DO NOT** contact mentors directly by email. Instead, please email [ml4-sci@cern.ch](mailto:ml4-sci@cern.ch) with Project Title. The relevant mentors will then get in touch with you.

## Submission Guidelines

- You are required to submit Jupyter Notebooks for each task clearly showing your implementation.
- Please also put your solution in a github repository and send us a link
- You must calculate and present the required evaluation metrics for the validation data (90:10 train-test split).
- When completed, please send your **CV**, a link to your **solution repository**, your **notebooks** and **trained model weights** to [ml4-sci@cern.ch](mailto:ml4-sci@cern.ch) with **Evaluation Test: DeepLense** in the title
- **Submission deadline is April 1**

## Common Test I. Multi-Class Classification

**Task:** Build a model for classifying the images into lenses using **PyTorch** or **Keras**. Pick the most appropriate approach and discuss your strategy.

**Dataset:** [dataset.zip - Google Drive](#)

**Dataset Description:** The Dataset consists of three classes, strong lensing images with no substructure, subhalo substructure, and vortex substructure. The images have been

normalized using min-max normalization, but you are free to use any normalization or data augmentation methods to improve your results.

**Evaluation Metrics:** ROC curve (Receiver Operating Characteristic curve) and AUC score (Area Under the ROC Curve)

- Those interested in the **Gravitational Lens Finding** Project should complete **Test II**
- Those interested in the **Data Processing Pipeline** Project should complete **Test II**
- Those interested in the **Super-resolution** Project should complete **Tests III.A and III.B**
- Those interested in the **Diffusion Models** Project should complete **Test IV**
- Those interested in the **Physics-Guided ML** Project should complete **Test V**
- Those interested in the **Foundation Model** Project should complete **Test VI**

## Specific Test II. Lens Finding

**Task:** Build a model identifying lenses using **PyTorch** or **Keras**. For the training use the images in `train_lenses` and `train_nonlenses` directories, for evaluation use the images from `test_lenses` and `test_nonlenses` directories. Note that the number of non-lenses is much larger than the number of lensed galaxies. Pick the most appropriate approach and discuss your strategy.

**Dataset:**

[https://drive.google.com/file/d/1doUhVoq1-c9pamZVLpvjW1YRDMkKO1Q5/view?usp=drive\\_link](https://drive.google.com/file/d/1doUhVoq1-c9pamZVLpvjW1YRDMkKO1Q5/view?usp=drive_link)

**Dataset Description:** A dataset comprising observational data of strong lenses and non-lensed galaxies. Images in [three different filters](#) are available for each object, so the shape of each object array is (3, 64, 64). Lensed objects are placed in the directory `train_lenses` and `test_lenses`, non-lensed galaxies are in `train_nonlenses` and `test_nonlenses`.

**Evaluation Metrics:** ROC curve (Receiver Operating Characteristic curve) and AUC score (Area Under the ROC Curve)

### Specific Test III. Image Super-resolution

**Task III.A:** Train a deep learning-based super resolution algorithm of your choice to upscale low-resolution strong lensing images using the provided high-resolution samples as ground truths. Please implement your approach in **PyTorch** or **Keras** and discuss your strategy.

**Dataset:**

[https://drive.google.com/file/d/1uJmDZw649XS-r-dYs9WD-OPwF\\_TlroVw/view?usp=sharing](https://drive.google.com/file/d/1uJmDZw649XS-r-dYs9WD-OPwF_TlroVw/view?usp=sharing)

**Dataset Description:** The dataset comprises simulated strong lensing images with no substructure at multiple resolutions: high-resolution (HR) and low-resolution (LR).

**Evaluation Metrics:** MSE (Mean Squared Error), SSIM (Structural Similarity Index), PSNR (Peak Signal-to-Noise Ratio)

**Task III.B:** Train a deep learning-based super-resolution algorithm of your choice to enhance low-resolution strong lensing images using a limited dataset of real HR/LR pairs collected from HSC and HST telescopes. You can adapt and fine-tune your super-resolution model from Task III.A. or use any other approach, such as few-shot learning strategies, transfer learning, domain adaptation, or data augmentation techniques, etc. Please implement your approach in **PyTorch** or **Keras** and discuss your strategy.

**Dataset:**

<https://drive.google.com/file/d/1pLYfM-jFJT7TbTMVssuCCFvLzGdxMQ4h/view?usp=sharing>

**Dataset Description:** The dataset comprises 300 strong lensing image pairs at multiple resolutions: high-resolution (HR) and low-resolution (LR).

**Evaluation Metrics:** MSE (Mean Squared Error), SSIM (Structural Similarity Index), PSNR (Peak Signal-to-Noise Ratio)

### Specific Test IV. Diffusion Models

**Task:** Develop a generative model to simulate realistic strong gravitational lensing images. Train a diffusion model ([DDPM](#)) to generate lensing images. You are

encouraged to explore various architectures and implementations within the diffusion model framework. Please implement your approach in **PyTorch** or **Keras** and discuss your strategy.

**Dataset:**

<https://drive.google.com/file/d/1cJyPQzVOzsCZQctNBuHCqxHnOY7v7UiA/view?usp=sharing>

**Dataset Description:** The dataset comprises 10,000 strong lensing images to train your model.

**Evaluation Metrics:** Use qualitative assessments and quantitative metrics such as Fréchet Inception Distance (FID) to measure the realism and variety of the generated samples.

## **Specific Test V. Physics-Guided ML**

**Task:** Build a model for classifying the images into lenses using **PyTorch** or **Keras**. Your architecture should take the form of a physics informed neural network ([PINN](#)). In this case, use the gravitational lensing equation in your architecture to improve network performance over your Common Test result.

**Dataset:** [dataset.zip - Google Drive](#)

**Dataset Description:** The Dataset consists of three classes, strong lensing images with no substructure, subhalo substructure, and vortex substructure. The images have been normalized using min-max normalization, but you are free to use any normalization or data augmentation methods to improve your results.

**Evaluation Metrics:** ROC curve (Receiver Operating Characteristic curve) and AUC score (Area Under the ROC Curve)

## **Specific Test VI. Foundation Model**

**Task VI.A:** Train a [Masked Autoencoder \(MAE\)](#) on the no\_sub samples from the provided dataset to learn a feature representation of strong lensing images. The MAE should be trained for reconstructing masked portions of input images. Once this pre-training phase is complete, fine-tune the model on the full dataset for a multi-class

classification task to distinguish between the three classes. Please implement your approach in **PyTorch** or **Keras** and discuss your strategy.

**Dataset:**

<https://drive.google.com/file/d/1znqUeFzYz-DeAE3dYXD17qoMPK82Whji/view?usp=sharing>

**Dataset Description:** The Dataset consists of three classes: no\_sub (no substructure), cdm (cold dark matter substructure), and axion (axion-like particle substructure).

**Evaluation Metrics:** ROC curve (Receiver Operating Characteristic curve) and AUC score (Area Under the ROC Curve)

**Task VI.B:** Take the pre-trained model from Task VI.A and fine-tune it for a super-resolution task. The model should be fine-tuned to upscale low-resolution strong lensing images using the provided high-resolution samples as ground truths. Please implement your approach in **PyTorch** or **Keras** and discuss your strategy.

**Dataset:**

[https://drive.google.com/file/d/1uJmDZw649XS-r-dYs9WD-OPwF\\_TlroVw/view?usp=sharing](https://drive.google.com/file/d/1uJmDZw649XS-r-dYs9WD-OPwF_TlroVw/view?usp=sharing)

**Dataset Description:** The dataset comprises simulated strong lensing images with no substructure at multiple resolutions: high-resolution (HR) and low-resolution (LR).

**Evaluation Metrics:** MSE (Mean Squared Error), SSIM (Structural Similarity Index), PSNR (Peak Signal-to-Noise Ratio)

## Submission Guidelines

- You are required to submit Jupyter Notebooks for each task clearly showing your implementation.
- Please also put your solution in a github repository and send us a link
- You must calculate and present the required evaluation metrics for the validation data (90:10 train-test split).
- When completed, please send your **CV**, a link to your **solution repository**, your **notebooks** and **trained model weights** to [ml4-sci@cern.ch](mailto:ml4-sci@cern.ch) with **Evaluation Test: DeepLense** in the title
- **Submission deadline is April 1**

