Dark Matter Substructure Classification
Using Simulated Lensing Images

Abstract

This project aims to classify dark matter through strong lensing images using different deep-learning frameworks and compare their performance. Images from DeepLense's simulated works are used to train the models. Our results highlight EfficientNet_B0 with the best accuracy and AUC, while the physics-informed Lensformer has the lowest loss. The project also emphasizes the advantages of CNNs and transformers when given more data and training time compared to our previous demo.

1/ Background & Theory:

Strong lensing images are astronomical images taken from telescopes that capture a phenomenon called strong gravitational lensing. Strong gravitational lensing is the intense bending of light due to large gravitational fields, such that the image of the source appears distorted or duplicated. These fields bend space-time around the object, causing the light to be deflected and travel in a "bent" path. The angle of deflection, α, is given by

$$\vec{\alpha} = \frac{2}{c^2}\frac{D_{LS}}{D_S D_L}\vec{\nabla}\int \mathrm{d}z\Psi(\vec{r}),$$

where D_{LS}, D_S, and D_L are the angular distances from the lens to the source, the observer to the source, and the observer to the lens, respectively. This equation assumes the lens to be thin, simplifying the equation. Additionally, $\psi(r)$ is the gravitational potential, which can be calculated if the angle of deflection and the respective distances are known. The gravitational potential can also be used to calculate the mass amount in the central galaxy (or other large object). Therefore, the proportion of lensing can be correlated to the total mass using the equation above. However, most of the mass of galaxies is not visible - it is dark matter. By analyzing strong lensing, the quantity and type of dark matter from the object can be calculated.

Dark matter has been a topic of interest for years. It does not interact with light, so it is tough to observe and measure. However, strong lensing provides a way to analyze dark matter. The shape and structure of the lensed image can give information about the kind of dark matter present in the object.

While strong lensing provides a good way of analyzing dark matter, only some clear images exist because it is rare and hard for telescopes to observe accurately. This means that computer simulations are usually needed to analyze strong lensing images.

2/ Data Description & Visualization: (train dataset, test dataset, simulation on GitHub)

Our dataset contains simulated images using lenstronomy, a Python library for modeling strong gravitational lenses. Images are evenly distributed among three classes: 'no_sub', 'cdm', and 'axion' for no substructure dark matter, cold dark matter, and axion dark matter, respectively. They are single-channel NPY files of 64x64 pixels, modeled with a simple Sersic light profile

and approximate an HST survey. This is the Model III dataset of DeepLense, which is more difficult to generalize than Model I in our demo.

In the training dataset, there are about 30,000 images per class, and 5,000 per class for the test dataset. During preprocessing, we noticed that for each axion dark matter image, there is an extra value corresponding to the axion's mass used in the simulation. Therefore, we only take the first element in these NPY files. After that, we created a validation set from the training set with a 90:10 split ratio. Here is some basic information about our processed data:

- Train dataset size: 80,457,
- Validation dataset size: 8,939,
- Test dataset size: 15,000,
- Batch size, channel, height, width: [100, 1, 64, 64],
- Image tensor type: torch.float32,
- Number of batches: 805.

Figure 1 shows 16 example images with their true labels. The Einstein rings, the prominent features of strong gravitational lensing images, are clearly displayed in each image. Some are complete, some are not (these are simulations, but a complete Einstein ring has never been observed before). Although the impact of different dark matter identities is not subtle on the ring, it is difficult to distinguish between classes in these simulated images.

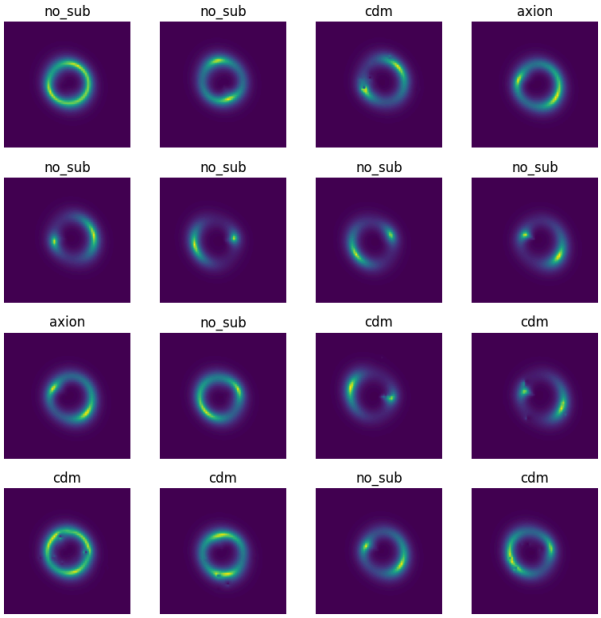

Figure 1. Visualization of data.

3/ Models & Methods:

Based on the recent studies in the PH 451 course, the group plans to use a standard CNN to classify the images. These results will be compared with four deep-learning models for computer vision mentioned in class: Resnet18, EfficientNet_B0, Vision Transformer,

Convolutional Transformer, and Lensformer. The standard CNN will be similar to the sample provided in our Mini-Hackathon #2. For the other architectures, there are a lot more parameters since the number of parameters takes a long time to adjust and optimize to make the benchmark "fair", the simplest version of each model type, which is already pre-defined in PyTorch's torchvision.models package, was used. This is the total number of trainable parameters for each model:

- CNN: 0.51 million,
- ResNet18: 11.17 million,
- EfficientNet_B0: 4.01 million,
- Vision Transformer: 13.40 million,
- Convolutional Transformer: 13.09 million,
- Lensformer: 13.24 million.

Each model will be trained for 100 epochs, using Cross Entropy Loss, Adam optimizer with a learning rate of 1e-5 (1e-4 for CNN and 1e-6 for CvT), and the ExponentialLR scheduler, which decays the learning rate by a factor of 0.8 every epoch.

The measurement of success for the project is the area-under-the-curve metric (AUC) to test the ability of the models to classify the lensing images into three categories. The AUC metric is commonly used to measure the performance and success of machine learning models. The metric itself is based on the receiver operating characteristic (ROC) curve, plotting the true positive rate (TPR) versus the false positive rate (FPR). The FPR represents the ratio of negative instances incorrectly classified as positive and equal to one minus the true negative rate. In the ROC curve, the higher the model recall (the TPR), the more false positives the model produces. A dotted line representing the ROC curve of a purely random classifier is generally included in the graph. Regarding the AUC metric, a perfect classifier will have a ROC AUC equal to 1. Specifically, one desires the curve to be closest to the top-left corner of the ROC graph to increase the AUC. A perfectly random classifier will have an ROC AUC equal to 0.5. An example ROC graph from the *Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow* book is included below.

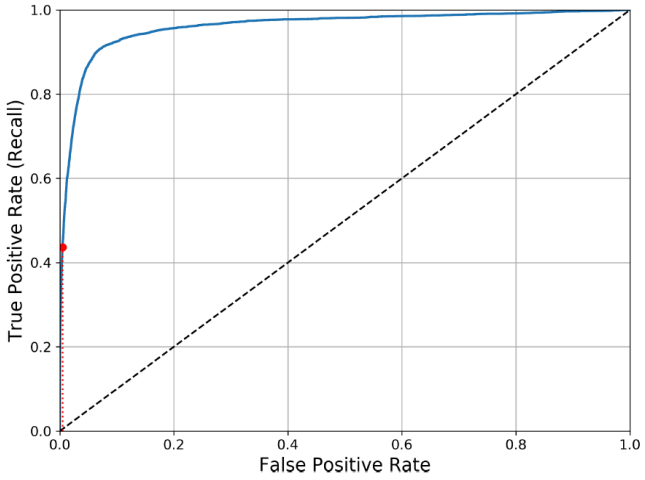

Figure 2. An example of a ROC curve

In the past, the AUC metric has been used for multiclass substructure classification of simulated dark matter substructure. For instance, the research paper "Deep Learning the Morphology of Dark Matter Substructure" by Stephon Alexander, Evan McDonough, and Sergei Gleyzer studied the capability of deep learning methods such as ResNet18 to classify the substructure of "dark matter based on simulated strong lensing images." An example of the resulting ROC graph illustrating the AUC values is shown below for the different classification categories of dark matter substructures, illustrating higher scores for dark matter with no simulated substructure. For the course project, the group intends to produce a similar ROC graph illustrating the final AUC scores. Compared to the graph shown below, the final graph produced for the project will contain three categories for dark matter classification and the model's macro average. One ROC curve will demonstrate the confidence level of every model used.

4/ Analysis & Results: (more details can be found in this PDF)

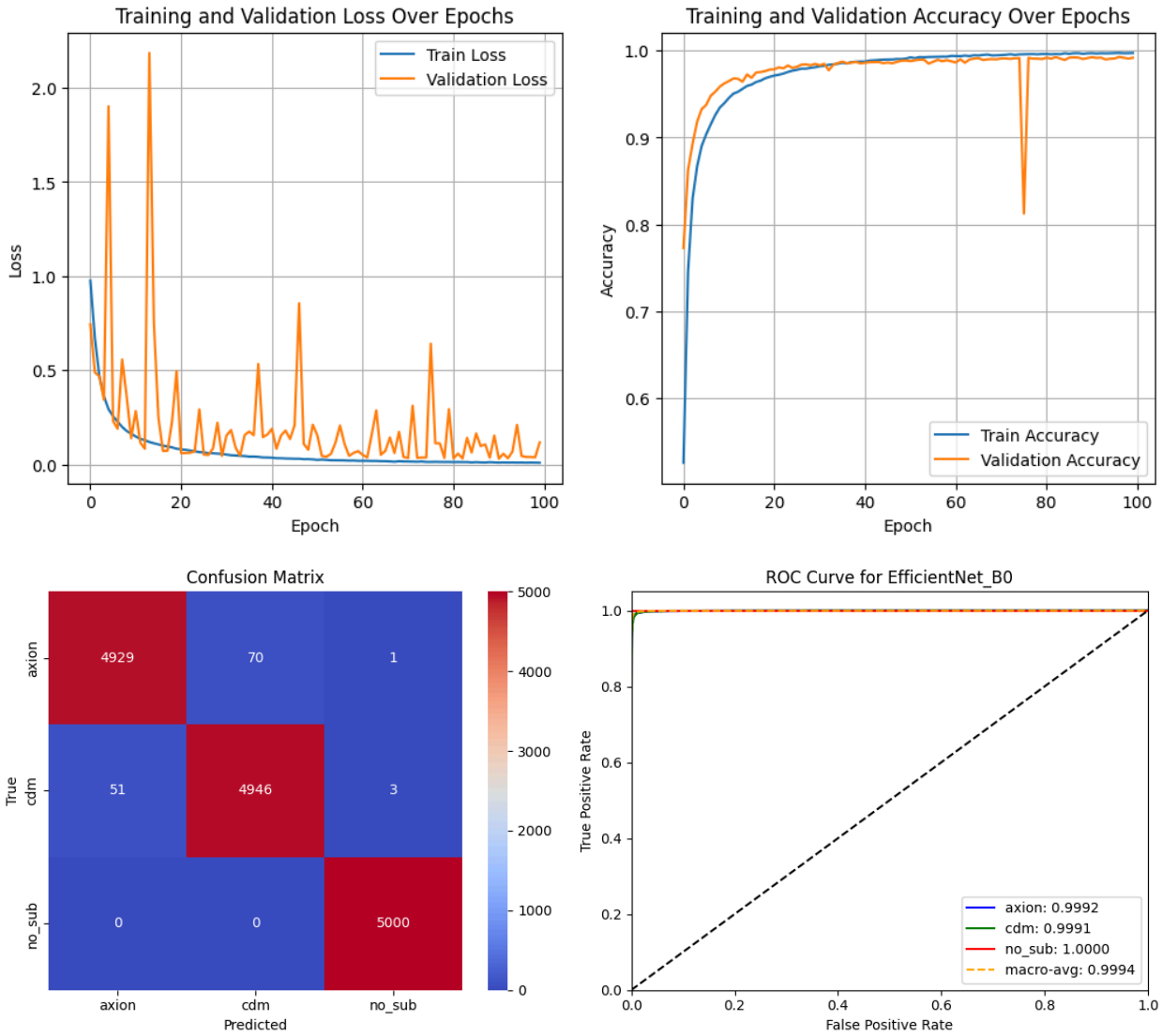

Figure 3. EfficientNet_B0 has the best results

	No. Params	Test Loss	Test Accuracy	ROC AUC
CNN	**0.51M**	0.1161	0.9555	0.9956
ResNet18	11.17M	0.0658	0.9828	0.9989
EfficientNet_B0	4.01M	0.0791	**0.9917**	**0.9994**
ViT	13.4M	0.1392	0.9525	0.9984
CvT	13.09M	0.0624	0.9785	0.9985
Lensformer	13.24M	**0.0446**	0.9845	0.9991

Table 1. Performance on test set

After 100 epochs, we can see that EfficientNet_B0 performs best, with 99.17% accuracy and 0.9994 ROC AUC. However, if we observe the loss per epoch graph, EfficientNet's loss fluctuates drastically from the start and becomes more stabilized later; there is also an abnormal drop in accuracy during the process. These fluctuations can be explained by a high learning rate, causing gradient descent to enter and exit the global minimum often. Another significance is Lensformer, which has the lowest test loss, 0.0446, indicating that it's more stable. The basic CNN with 0.51 million parameters is the most efficient regarding training time.

The figures below demonstrate the signals that traditional CNN and ResNet18 picked up after the first convolution (EfficientNet and CvT have more complex architecture, making their outputs nearly or entirely black):

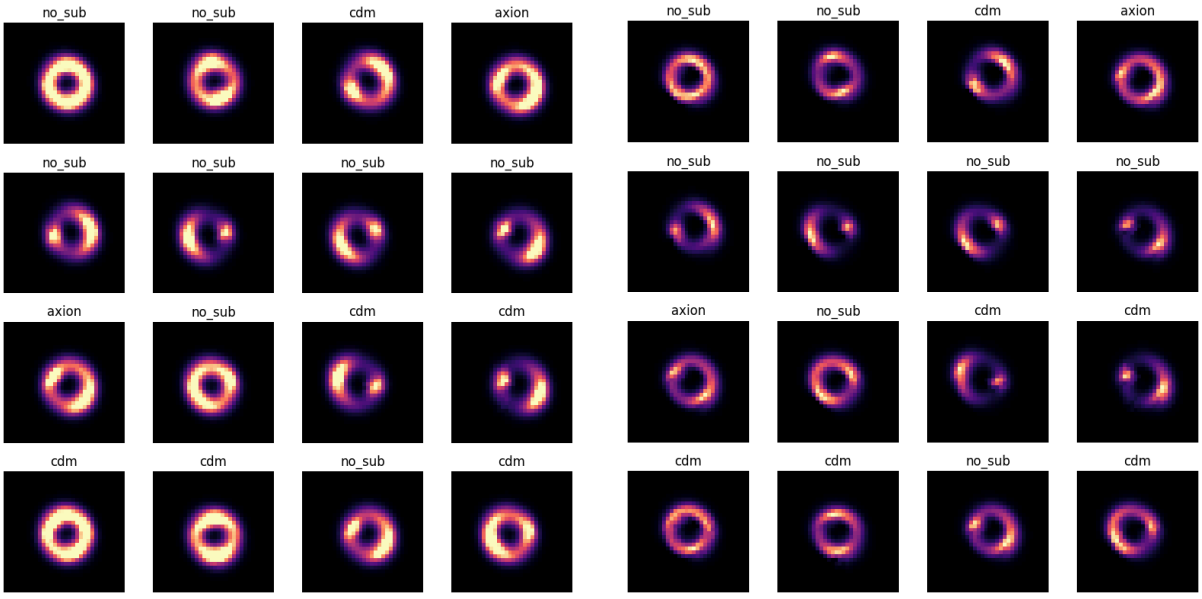

Figure 4. Outputs after 1st convolution of CNN (left) and ResNet18 (right)

These are the same images from the visualizations in Figure 1. Both convolutional models immediately ignored the background and focused on the features of the Einstein rings; the only difference is that the pixels from CNN's outputs are brighter and spread out compared to those from ResNet18. In the case of EfficientNet and CvT, their first convolutions are usually meant for upsampling, so unless we visualize what happens after the residual bottleneck block for EfficientNet and patch tokenization for CvT, we would get pitch-black images with no clear features.

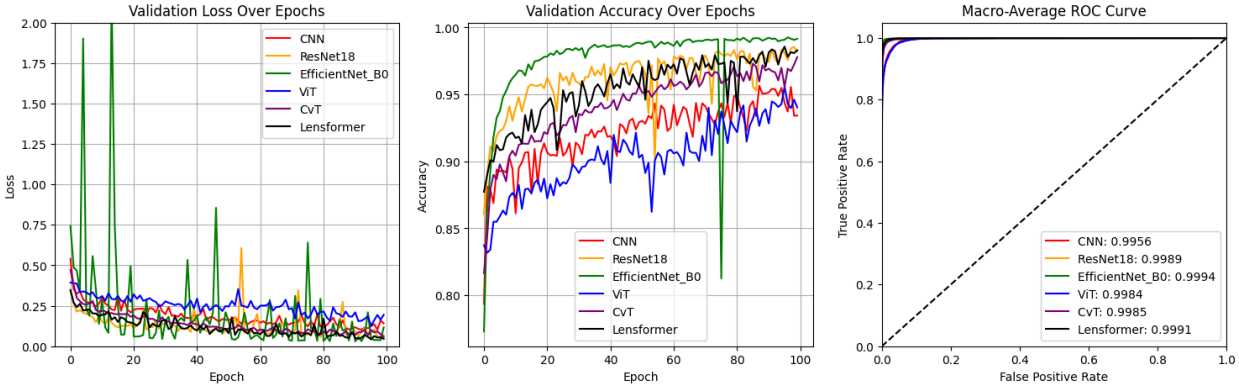

Figure 5. Validation loss, accuracy, and ROC curves of all models

If we look carefully at both loss and accuracy over epochs, it is evident that Lensformer has had the best loss and accuracy from the beginning. This is significantly due to the physics-informed features in its architecture, which enable superior generalization. The other convolutional architectures like CNN, ResNet18, EfficientNet, and CvT can leverage local invariance and inductive bias to make early assumptions, especially ResNet18 and EfficientNet, which have many convolutions. ViT, which heavily relies on training time and data volume, has the worst early run out of all models. However, given more training time and data, all transformer architectures should outperform the CNNs, as previous studies pointed out.

Looking at the ROC curves, all models have excellent scores above 0.99 AUC. This is surprising since our demo cannot achieve this performance with DeepLense's Model I, which is supposed to be simpler simulations. Even at the 20 epochs checkpoint, the results from Model III are far better. Last but not least, the models' performance in each dark matter class is similar in this order: 'no_sub', 'axion', then 'cdm' from best to worst. Because dark matter without substructure has been ruled out by physics, we suspect this class has the most minor correlation with the others, hence the highest AUC. We also know that axion is a form of cold dark matter; this can explain why cold dark matter usually has the lowest AUC score.

5/ Conclusion:

Deep learning techniques can achieve higher than 90% accuracy in classifying dark matter substructures through gravitational lensing images, promising excellent efficiency in future studies. The project highlights are EfficientNet_B0 and Lensformer. With only 4.01M parameters, EfficientNet yielded the best performance, given that its training process was volatile

compared to other models. The learning rate 1e-5 gave this model the best accuracy of 99.17% but occasionally overshoots the global minimum. On the other hand, Lensformer is a more reliable option with the lowest cross-entropy loss of 0.0446 on the test set. There are also slight differences between training and validation during training, indicating great capability to generalize on unseen data. Although deep learning techniques almost perfectly classify dark matter substructure through lensing simulations, we must anticipate a drop in performance when applying them to real images. These simulations are modeled after specific physic principles (simple Sersic light profile in our case), so they won't be able to capture all features from natural events.

We successfully addressed the issues with training time and the lack of data from our demo, which caused the transformers to underperform. Throughout this course's final project, we learned the subtleties of different models during the training process, such as the strengths and drawbacks of CNNs versus transformers in computer vision tasks. Our dataset has an obvious ring-like pattern, which convolutional networks can leverage its translation invariance and inductive bias to assume signals and converge at the global minimum faster. However, these advantages might not apply to a more complex dataset, which can cause these CNNs to make a wrong assumption and guess randomly. In this situation, the transformers are more reliable with their global context awareness and fewer invariance imposed when given enough data and epochs. Our experiments show that picking the suitable model depends on what kind of data you're dealing with and what you're trying to achieve. It'd be great to see more research on how these models deal with different challenges and more complicated, real-world data.

6/ References:

Alexander, S., Gleyzer, S., McDonough, E., Toomey, M. W., & Usai, E. (2020). Deep learning the morphology of dark matter substructure. The Astrophysical Journal, 893(1), 15.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.

Narayan, R., & Bartelmann, M. (1996). Lectures on gravitational lensing. arXiv preprint astro-ph/9606001.

Peccei, R. D., & Quinn, H. R. (1977). CP conservation in the presence of pseudoparticles. Physical Review Letters, 38(25), 1440.

Peccei, R. D. (1978). A short review of axions.

Tan, M., & Le, Q. (2019, May). Efficientnet: Rethinking model scaling for convolutional neural networks. In International conference on machine learning (pp. 6105-6114). PMLR.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.

Velôso, L. J., Toomey, M. W., & Gleyzer, S. Lensformer: A Physics-Informed Vision Transformer for Gravitational Lensing.

Wambsganss, J. (1998). Gravitational lensing in astronomy. Living Reviews in Relativity, 1, 1-74.

Wilczek, F. (2013). Superfluidity and space-time translation symmetry breaking. Physical review letters, 111(25), 250402.

Wu, H., Xiao, B., Codella, N., Liu, M., Dai, X., Yuan, L., & Zhang, L. (2021). Cvt: Introducing convolutions to vision transformers. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 22-31).