Extension of Source Code Package for Open-Set Recognition (OSR)

Team: TBD

Supervisor: Laurin van den Bergh, Prof. Dr. Manuel Günther

Intended Start: Fall 2024

1 Introduction

The automatic separation of images into several classes has gained a lot of interest in the last years. Starting with small-scale datasets such as MNIST and CIFAR-10, nowadays more realistic and large-scale datasets are used for this task. Particularly, the availability of the ImageNet dataset [Deng et al., 2009] and its usage in the International Large Scale Visual Recognition Challenge (ILSVRC) [Russakovsky et al., 2015] has fostered large improvements in this task. Especially, deep neural networks have shown great success in the ImageNet challenge [Krizhevsky et al., 2012, He et al., 2015].

However, most classifiers have not left academic areas since they have one important flaw: these classifiers can only classify samples of classes that they have seen during training. When presented with a sample from a different class that the network is not trained to predict, it has no other chance than predicting one of the known classes, which it oftentimes does with large confidence [Dhamija et al., 2018, Dhamija et al., 2020, Palechor et al., 2023].

Lately, researchers have understood this limitation and provided different ways of providing options for the classifier to reject such samples as unknown. There are three main directions of such approaches. The first approaches, which we call post-processing approaches, are taking a pre-trained network that is trained on a closed-set task such as ImageNet, and provide means and options to reject a sample. For example [Hendrycks and Gimpel, 2017, Hendrycks et al., 2022] showed that thresholding the softmax probabilities, or the logits that feed into softmax, are good baselines for open-set recognition. Other approaches try to approximate the probability of unknown by modeling deep feature representations [Bendale and Boult, 2016, Rudd et al., 2017, Lyu et al., 2023] of known classes, to provide a probability of sample exclusion.

The second main direction tries to train the networks such that they are better suited for open-set recognition. Most approaches in this direction add another output for the unknown class to the network, while others include different loss functions to enable better thresholding of outputs [Dhamija et al., 2018]. To obtain training samples for the unknown class (which we term *negative* samples), different methods are developed. In the simplest way, negative samples are collected from classes that are not of interest [Dhamija et al., 2018, Palechor et al., 2023]. Other approaches artificially generate such negative samples in various different ways [Ge et al., 2017, Yu et al., 2017, Neal et al., 2018, Wilson et al., 2023, Geng et al., 2021]. For example, mixed representations of known classes are used as negatives [Zhou et al., 2021], with the hope that they are good predictors for unknown samples.

The third type of methods make use of a two-stage process. First the task is to classify whether a sample belongs to the distribution of known samples via Out-of-Distribution Detection methods [Cui and Wang, 2022] – several methods for this task are implemented in the OpenOOD package [Yang et al., 2022][1]. These scores are then combined with the closed-set classification (there is actually very little research on this combination, including [Lang, 2024]).

Finally, the validation and evaluation of open-set recognition approaches make use of various methods, for example, computing AUROC for the binary out-of-distribution detection task, or computing the Open-Set Classification Rate (OSCR) curve [Dhamija et al., 2018].

Most of the above algorithms are evaluated on small-scale datasets with particular properties, which do not reflect real-world applications well. Since there was a lack of large-scale evaluation protocols for open-set recognition, we have developed our own evaluation [Palechor et al., 2023] based on the ImageNet dataset. In that paper, we compared simple approaches for open-set recognition, for which we also have published an

[1] `https://github.com/Jingkang50/OpenOOD`

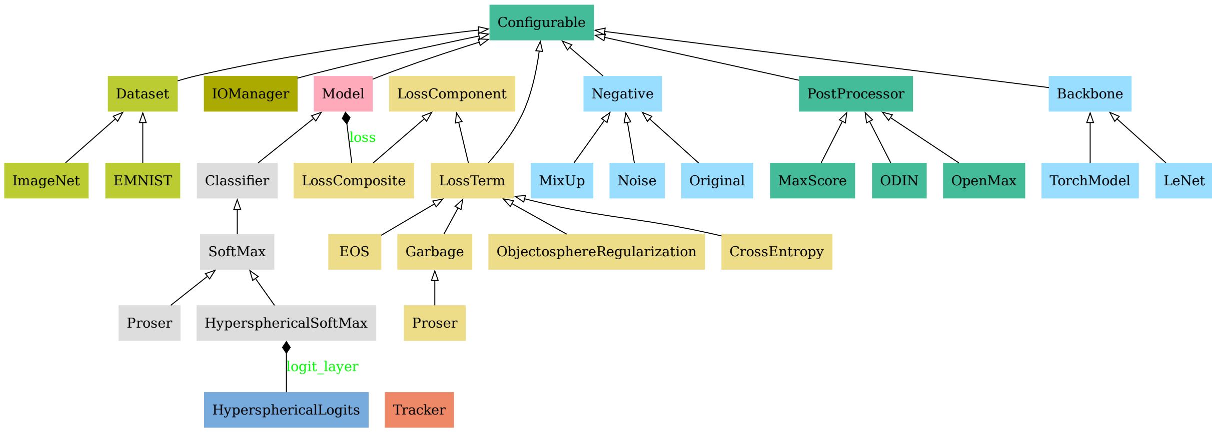


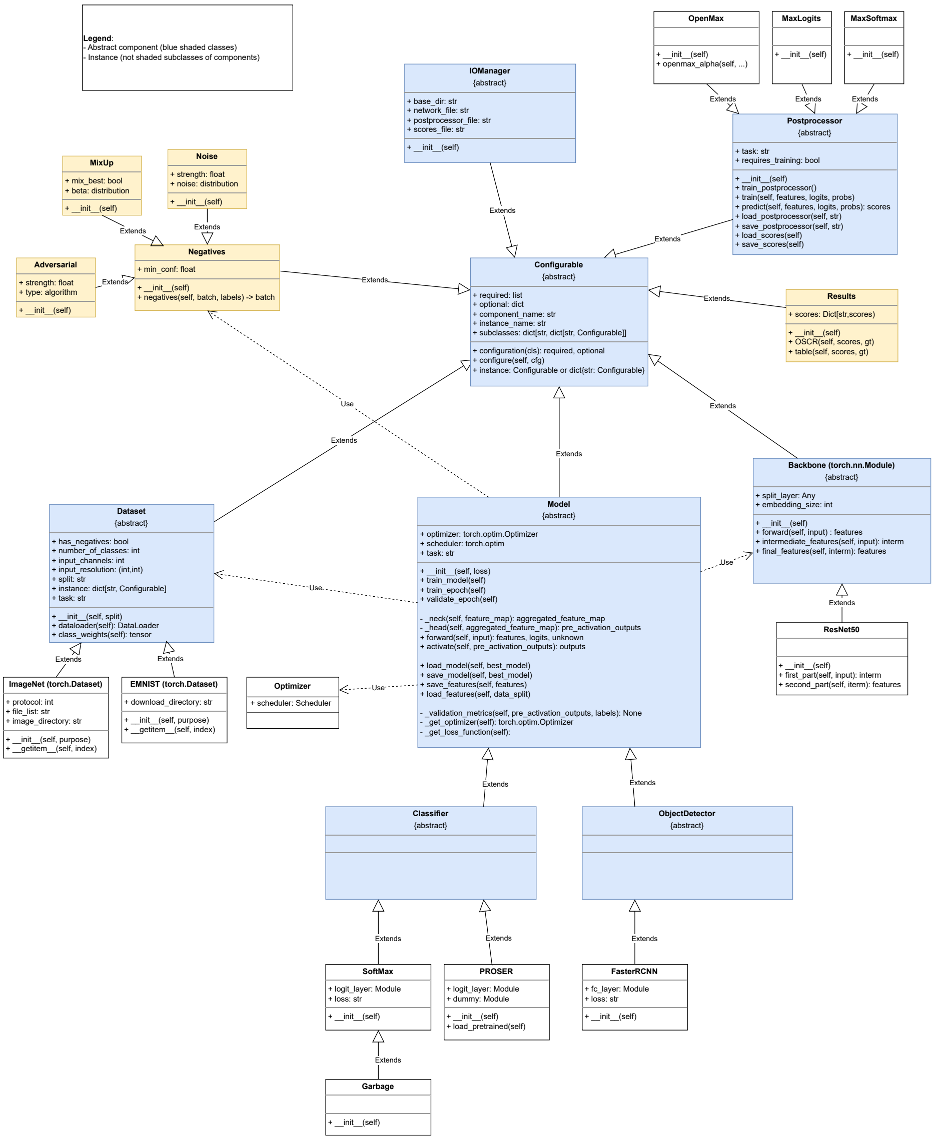
Figure 1: Currently implemented components and methods in the OpenOSR package.

open-source code package.[2] Currently, we are extending this paper to include more algorithms to compare[3] – already implemented methods include [Dhamija et al., 2018, Zhou et al., 2021, Hendrycks and Gimpel, 2017, Hendrycks et al., 2022, Bendale and Boult, 2016, Rudd et al., 2017] (see Figure 1)

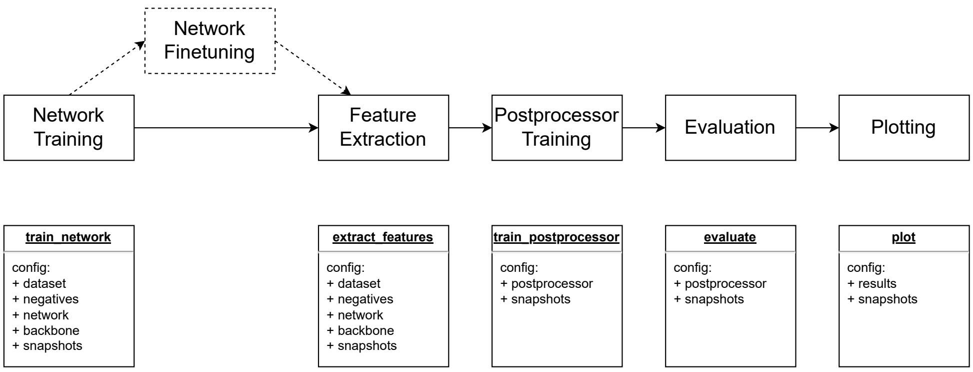
Finally, we started to design and implement a new open-source package, OpenOSR,[4] that combines various aspects of open-set recognition. We have introduced a complete design concept, which includes various base classes and derived implementations of various methods. A class and a flow diagram of the new package design can be found in Figure 2. We have implemented a few datasets, some basic algorithms for training and post-processing methods, and several validation and evaluation metrics. The task of this Master Project would be to extend this package by:

1. Familiarization:
   - Understand the design concept presented in our class diagram (Figure 2) and the code basis.
   - Read the current documentation.
   - Run simple small-scale experiments to test correct setup of the environment.
   - Document weaknesses in the user-friendliness of the current version and suggest improvements.
2. Documentation:
   - Design/reorganize the documentation to be more user-friendly and as concise as possible (this can include the reorganization of current pages, inclusion of a tutorial for users and developers, a quick start guide etc.).
   - Document the design concept in the Sphinx documentation and update the current version.
   - Make sure that all existing methods are documented, including all of their parameters.
   - Stretch: setup automatic online documentation generation with GitHub actions.
3. Implementations:
   - Implement some datasets used in the literature, including CIFAR-10/100, ...
   - Implement various strategies (models and postprocessors) for open-set recognition.
   - Implement more validation and evaluation metrics.
   - Provide documentation for each implemented method.
4. Testing:
   - Implement test cases for all implemented methods.

[2] `https://github.com/AIML-IfI/openset-imagenet`
[3] `https://github.com/AIML-IfI/openset-imagenet-comparison`
[4] `https://github.com/AIML-IfI/OpenOSR`



(a) Class Diagram



(b) Flow Diagram

Figure 2: Current versions of Class and Flow Diagrams of the Package.

- Stretch: set up continuous integration framework on GitHub to run test cases automatically, and run coverage tests.

5. Experiments:

   - Showcase the correctness of the implementations by running and evaluating newly implemented methods and datasets.
   - Combine or compare methods with previously implemented methods.

2 Schedule

The project is designed for two to four students. For two students, the focus should be on implementing, testing and documenting datasets and open-set metrics. When more students are present, the continuous integration framework should be implemented.

Assuming 30 hours of work per week and a total of 15 ECTS with an average of 30 hours per ECTS, we arrive at a total workload of 15 weeks full-time. These could be distributed as follows.

Week 1-3 Setting up the work environment, installing all required tools, (partially) reproduce the results that are currently available in the source code package. Understand the core concept, which is best achieved by writing documentation. Possibly write a tutorial on how to extend the framework.

⇒ Milestone 1: The documentation of the package is updated, and a tutorial is written.

Week 4-5 Perform a guided literature review and search for available open-source implementations of other datasets and algorithms. Algorithms do not need to be restricted to image-based classification, for example, approaches from other domains, e.g., [Wu et al., 2020, Zhan et al., 2021] can be included if possible.

⇒ Milestone 2: A collection of open-source datasets and methods is selected that should be included into the package.

Week 6-11 Extend the software package by following and refining the tutorial created in the first weeks. Implement the methods and datasets selected in Milestone 2, including test cases and documentation. Also, validation and evaluation metrics should be included.

⇒ Milestone 3: New open-set methods are incorporated into the package,

Week 12-15 Run experiments with the new methods and compare/combine them with other methods.

⇒ Milestone 4: Experiments have been run and plots have been generated.

If time allows Run experiments to optimize parameters of the implemented algorithms to get the best possible result on the protocols.

Optionally Set up continuous integration and automatic online documentation generation via GitHub actions.

Writing the project report is part of the Master project. As a template, the LaTeX thesis template from my webpage[5] should be used. I would recommend to start writing early and keep note of what was done when, and by whom. At the end of the project, there will be a joint presentation of the results in my research group.

3 References

[Bendale and Boult, 2016] Bendale, A. and Boult, T. E. (2016). Towards open set deep networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

[Cui and Wang, 2022] Cui, P. and Wang, J. (2022). Out-of-distribution (OOD) detection based on deep learning: A review. *Electronics*, 11(21).

[Deng et al., 2009] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

[5] `https://www.ifi.uzh.ch/en/aiml/theses.html`

[Dhamija et al., 2020] Dhamija, A., Günther, M., Ventura, J., and Boult, T. E. (2020). The overlooked elephant of object detection: Open set. In *Winter Conference on Applications of Computer Vision (WACV).*

[Dhamija et al., 2018] Dhamija, A. R., Günther, M., and Boult, T. E. (2018). Reducing network agnostophobia. In *Advances in Neural Information Processing Systems (NeurIPS).*

[Ge et al., 2017] Ge, Z., Demyanov, S., and Garnavi, R. (2017). Generative OpenMax for multi-class open set classification. In *British Machine Vision Conference (BMVC).*

[Geng et al., 2021] Geng, C., Huang, S.-J., and Chen, S. (2021). Recent advances in open set recognition: A survey. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 43(10):3614–3631.

[He et al., 2015] He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *International Conference on Computer Vision (ICCV)*, pages 1026–1034.

[Hendrycks et al., 2022] Hendrycks, D., Basart, S., Mazeika, M., Zou, A., Kwon, J., Mostajabi, M., Steinhardt, J., and Song, D. (2022). Scaling out-of-distribution detection for real-world settings. In *International Conference on Machine Learning (ICML).* PMLR.

[Hendrycks and Gimpel, 2017] Hendrycks, D. and Gimpel, K. (2017). A baseline for detecting misclassified and out-of-distribution examples in neural networks. *International Conference on Learning Representations (ICLR).*

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NeurIPS).*

[Lang, 2024] Lang, D. (2024). Combined out-of-distribution detection and open-set classification. Master's thesis, University of Zurich.

[Lyu et al., 2023] Lyu, Z., Gutierrez, N. B., and Beksi, W. J. (2023). Metamax: Improved open-set deep neural networks via weibull calibration. In *Winter Conference on Applications of Computer Vision Workshops (WACVW).*

[Neal et al., 2018] Neal, L., Olson, M., Fern, X., Wong, W.-K., and Li, F. (2018). Open set learning with counterfactual images. In *European Conference on Computer Vision (ECCV).*

[Palechor et al., 2023] Palechor, A., Bhoumik, A., and Günther, M. (2023). Large-scale open-set classification protocols for ImageNet. In *Winter Conference on Applications of Computer Vision (WACV).* CVF/IEEE.

[Rudd et al., 2017] Rudd, E. M., Jain, L. P., Scheirer, W. J., and Boult, T. E. (2017). The extreme value machine. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI).*

[Russakovsky et al., 2015] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115(3).

[Wilson et al., 2023] Wilson, S., Fischer, T., Dayoub, F., Miller, D., and Sünderhauf, N. (2023). SAFE: Sensitivity-aware features for out-of-distribution object detection. *arXiv preprint arXiv:2208.13930.*

[Wu et al., 2020] Wu, M., Pan, S., and Zhu, X. (2020). OpenWGL: Open-world graph learning. In *International Conference on Data Mining (ICDM).*

[Yang et al., 2022] Yang, J., Wang, P., Zou, D., Zhou, Z., Ding, K., PENG, W., Wang, H., Chen, G., Li, B., Sun, Y., Du, X., Zhou, K., Zhang, W., Hendrycks, D., Li, Y., and Liu, Z. (2022). OpenOOD: Benchmarking generalized out-of-distribution detection. In *Advances in Neural Information Processing Systems.*

[Yu et al., 2017] Yu, Y., Qu, W.-Y., Li, N., and Guo, Z. (2017). Open-category classification by adversarial sample generation. In *International Joint Conference on Artificial Intelligence (IJCAI).*

[Zhan et al., 2021] Zhan, L., Liang, H., Liu, B., Fan, L., Wu, X., and Lam, A. Y. S. (2021). Out-of-scope intent detection with self-supervision and discriminative training. In *International Joint Conference on Natural Language Processing, (IJCNLP).* Association for Computational Linguistics.

[Zhou et al., 2021] Zhou, D.-W., Ye, H.-J., and Zhan, D.-C. (2021). Learning placeholders for open-set recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR).*