


ImageMLResearch: A Python Toolkit for Reproducible Image-Based ML Experiments

Luis Kraker¹ and Gudrun Schappacher-Tilp¹

¹ FH JOANNEUM University of Applied Sciences, Graz, Austria  Corresponding author

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

Software

- [Review](#)
- [Repository](#)
- [Archive](#)

Editor: 

Submitted: 09 October 2025

Published: unpublished

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/))

ImageMLResearch: A Python Toolkit for Reproducible Image-Based ML Experiments

ImageMLResearch is an open-source Python toolkit that streamlines and standardizes image-based machine learning (ML) research. While ML has achieved remarkable success in computer vision, the complexity of research workflows remains a barrier to reproducibility and accessibility. Many projects rely on loosely connected scripts or notebooks, leading to fragmented experiment management and limited reproducibility.

ImageMLResearch addresses this gap by providing a modular Python package with a clear API, without requiring intrusive dashboards or command-line interfaces. Built on widely adopted libraries such as TensorFlow (Abadi et al., 2016), Keras (Chollet & others, 2015), and Optuna (Akiba et al., 2019), it offers a lightweight, research-oriented approach to reproducible image-based ML experimentation. The toolkit is designed to support education, exploratory research, and the development of more robust experiment management practices.

Statement of Need

Image-based machine learning workflows are often constructed from ad hoc scripts or notebooks, making it difficult to maintain a clear structure between data handling, preprocessing, training, and evaluation. This fragmentation contributes to poor reproducibility and hinders systematic experimentation (Gundersen et al., 2018; Hutson, 2018; Pineau et al., 2021).

General-purpose frameworks such as TensorFlow (Abadi et al., 2016) and PyTorch (Paszke et al., 2019) provide the computational foundations but do not prescribe experiment organization. Tools such as MLflow (Zaharia et al., 2018) and Weights & Biases (Biewald, 2020) extend functionality with experiment tracking and visualization, yet they often require additional infrastructure that can be burdensome in lightweight academic projects.

ImageMLResearch addresses this gap by structuring the experiment lifecycle for image classification into modular components. It supports the consistent definition, execution, and documentation of experiments without reliance on external services. This makes the toolkit especially suited for exploratory research, and smaller projects where transparent and reproducible experiment management is essential.

Software Description

ImageMLResearch is implemented in Python and integrates TensorFlow, Keras, and Optuna. It provides five research modules:

- Data Handling** – for structured dataset loading and preparation
- Preprocessing** – for image normalization and augmentation
- Plotting** – for visualizing data distributions, training curves, and results

- **Training** – for orchestrating model construction and optimization
- **Experimenting** – for automated runs, logging, and evaluation

These modules are coordinated through high-level Researcher classes that integrate the experiment lifecycle. Assets are organized into **definition**, **execution**, and **output** layers, ensuring clear separation of concerns. The toolkit automatically tracks logs, figures, and experiment metadata, generating human-readable markdown reports. Hyperparameter optimization is supported through Optuna, and a proof-of-concept AI-assisted analysis feature demonstrates automated interpretation of experiment results.

Illustrative Example

The structure of an ImageMLResearch experiment is illustrated in the diagram below.

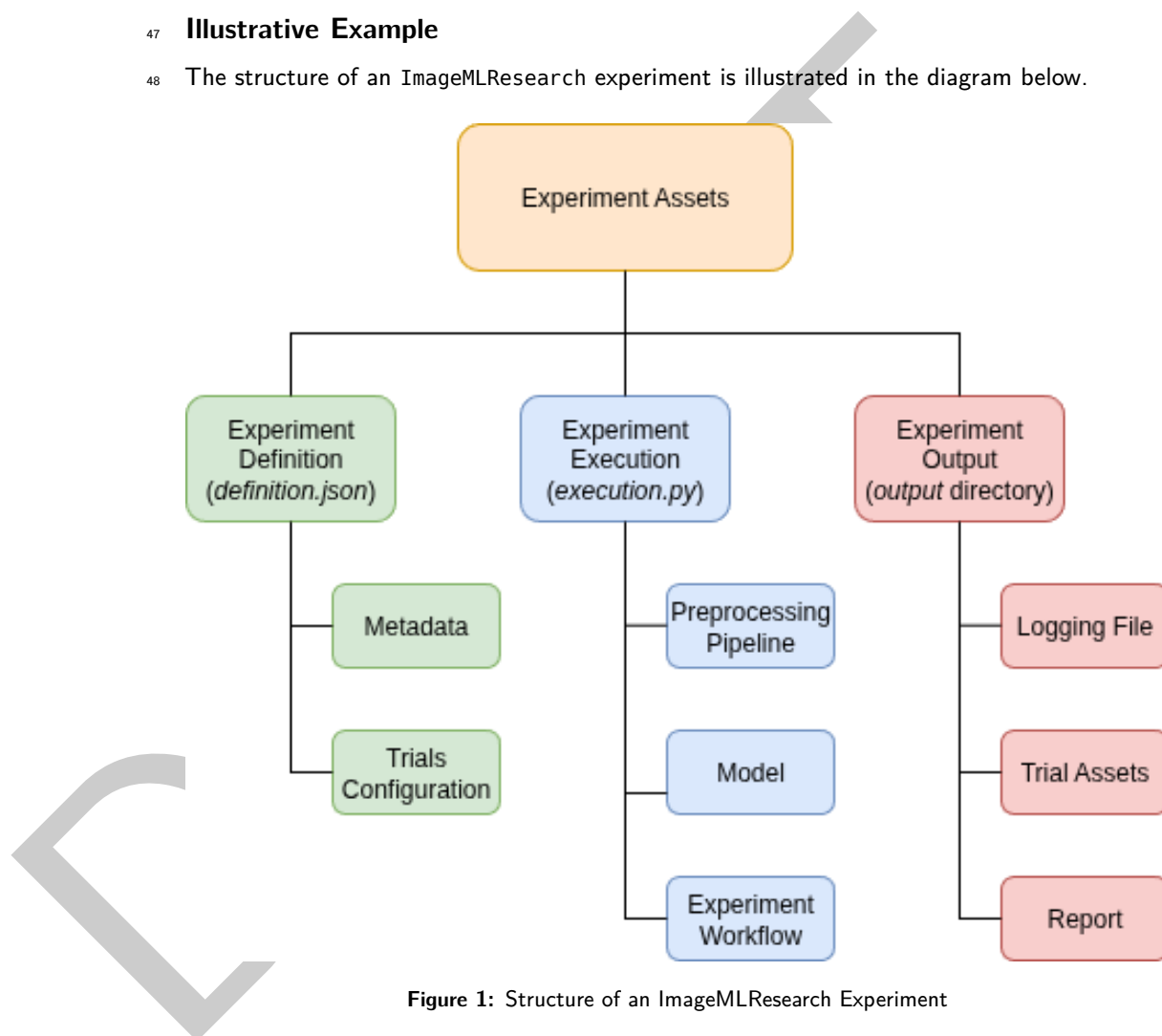


Figure 1: Structure of an ImageMLResearch Experiment

The metadata specifies the experiment name, directory, and sorting metric, while trials can be configured either manually or generated automatically through hyperparameter tuning. For example, running an MNIST digit experiment with two trials produces the following directory structure.

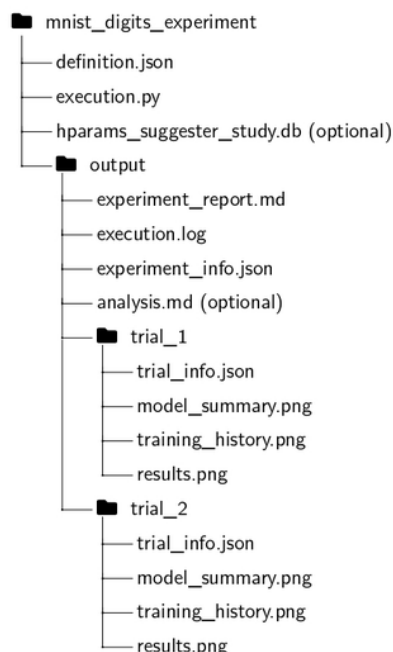


Figure 2: Output directory layout for a two-trial MNIST experiment

Quality Control

ImageMLResearch is maintained under version control with Git and GitHub. Unit tests are implemented with Python's unittest framework for each module, executed with a dedicated test runner that reports pass/fail/error logs. Code quality is enforced using Pylint and Ruff in accordance with PEP 8. AI-assisted consistency checks are performed with GitHub Copilot.

Acknowledgements

Developed under the FFG Coin ENDLESS Research Project.

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., & others. (2016). TensorFlow: Large-scale machine learning on heterogeneous distributed systems. *arXiv Preprint arXiv:1603.04467*.

Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2623–2631.

Biewald, L. (2020). *Experiment tracking with weights and biases*. Software available from <https://wandb.com>.

Chollet, F., & others. (2015). Keras. GitHub repository. <https://github.com/keras-team/keras>

Gundersen, O. E., Gil, Y., & Aha, D. W. (2018). On reproducible AI: Towards reproducible research, open science, and digital scholarship in AI publications. *AI Magazine*, 39(3), 56–68. <https://doi.org/10.1609/aimag.v39i3.2816>

Hutson, M. (2018). Artificial intelligence faces reproducibility crisis. *Science*, 359(6377), 725–726. <https://doi.org/10.1126/science.359.6377.725>

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., & others. (2019). PyTorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*,

77 32.

78 Pineau, J., Vincent-Lamarre, P., Sinha, K., Larivière, V., Beygelzimer, A., d'Alché-Buc, F.,
79 Fox, E., & Larochelle, H. (2021). Improving reproducibility in machine learning research
80 (a report from the NeurIPS 2019 reproducibility program). *Journal of Machine Learning*
81 *Research*, 22(164), 1–20.

82 Zaharia, M., Chen, A., Davidson, A., Ghodsi, A., Hong, S. A., Konwinski, A., Murching,
83 S., Nykodym, T., Ogilvie, M., Parkhe, M., & others. (2018). Accelerating the machine
84 learning lifecycle with MLflow. *Proceedings of the 4th International Workshop on Data*
85 *Management for End-to-End Machine Learning*, 39–44.

DRAFT