The Journal of Open Source Software

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

Luis Kraker [1¶] **and Gudrun Schappacher-Tilp** [1]

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

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Neea Rusch

Reviewers:

- @Amorfati123
- @sibocw

Submitted: 09 October 2025
Published: unpublished

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

Summary

`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).

While modern machine learning libraries provide powerful computational building blocks, they do not enforce a coherent structure for managing experiments. As a result, researchers must manually coordinate configurations, results, and documentation, which increases cognitive overhead and the likelihood of irreproducible outcomes.

`ImageMLResearch` was developed to address these challenges by providing a lightweight, structured framework for defining, executing, and documenting image-based machine learning experiments in a reproducible manner.

State of the Field

A variety of tools exist to support machine learning experimentation and reproducibility. Core frameworks such as TensorFlow (Abadi et al., 2016) and PyTorch (Paszke et al., 2019) provide flexible abstractions for model development and training but leave experiment organization and result management largely to the user.

Experiment tracking platforms such as MLflow (Zaharia et al., 2018) and Weights & Biases (Biewald, 2020) address this limitation by offering centralized logging, visualization dashboards, and metadata management. While powerful, these systems typically rely on external services and introduce additional infrastructure and configuration overhead, which can be a barrier in lightweight academic or educational settings.

In contrast, `ImageMLResearch` focuses on structuring the full experiment lifecycle for image-based machine learning within a self-contained Python package. Rather than emphasizing dashboards or large-scale tracking, it prioritizes transparent configuration, deterministic experiment definitions, and file-based artifacts tailored to image data. This positions the toolkit between low-level ML frameworks and full-scale experiment management platforms, addressing the needs of reproducible, small-to-medium-scale image-based research projects.

Software Design

`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.

The software design emphasizes reproducibility through explicit configuration and deterministic experiment definitions, and portability through file-based outputs rather than reliance on external services. The modular structure allows individual components (e.g., preprocessing or training strategies) to be replaced without changing the surrounding experiment orchestration, supporting method comparison and benchmarking with minimal boilerplate.

Research Impact Statement

ImageMLResearch is designed to lower the barrier to systematic experimentation in academic and educational settings. By standardizing workflows from data preparation to reporting, the toolkit allows researchers to focus on hypothesis-driven investigation rather than infrastructure maintenance.

In research contexts, the software supports rigorous benchmarking and method comparison, which are essential for reproducible and peer-reviewed machine learning studies. ImageMLResearch was used within the FFG-funded ENDLESS research project to ensure that complex image-classification experiments remained reproducible across collaborating research teams.

In educational settings, the toolkit provides a structured framework for teaching best practices in machine learning experimentation. By enforcing a clear separation between experimental definitions and generated outputs, it encourages students to approach machine learning experiments as structured scientific studies rather than collections of disconnected trial-and-error scripts.

AI Usage Disclosure

OpenAI's ChatGPT was used to enhance clarity and readability of the manuscript. AI-assisted code completion and consistency checks were performed using GitHub Copilot during software

The Journal of Open Source Software

development. All AI-generated suggestions were reviewed, verified, and edited by the authors to ensure correctness and scientific accuracy.

The authors maintain full responsibility for the software's architecture, the implementation of the core research logic, and the scientific validity of the experimental results. All AI-suggested content was manually audited, refined, and verified to ensure it meets the rigorous standards of research software. No core algorithmic logic or novel research methodology was generated by AI.

Illustrative Example

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

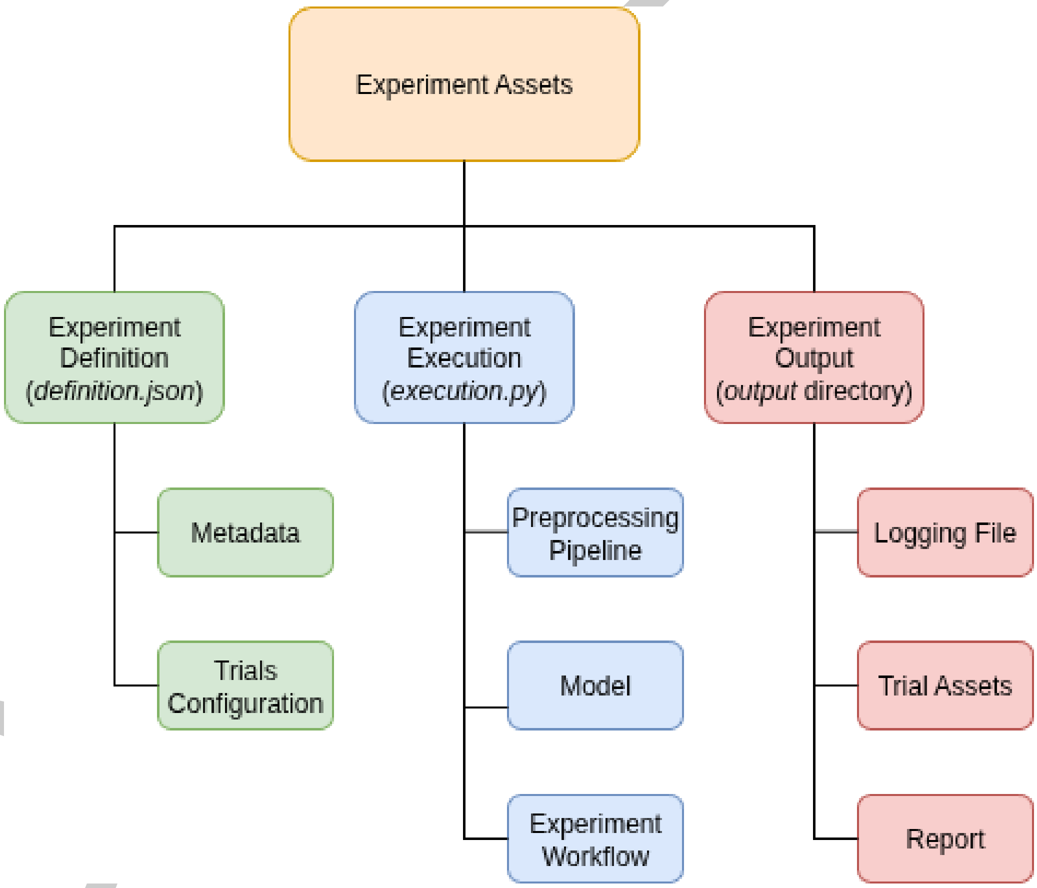

Figure 1: Structure of an ImageMLResearch Experiment

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.

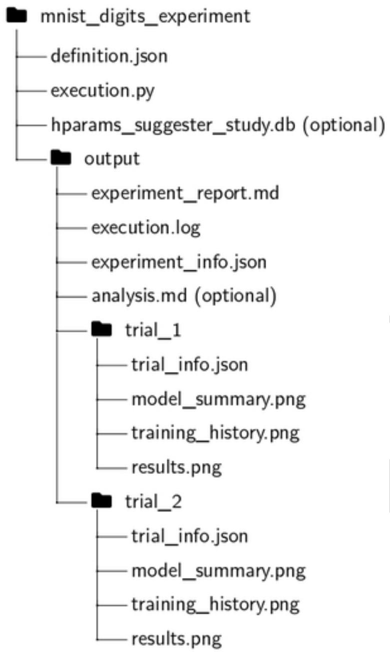

Figure 2: Output directory layout for a two-trial MNIST experiment

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*, *32*.

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

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