MISeval: A Metric Library for Medical Image Segmentation Evaluation

Dominik MÜLLER[a,b,1], Dennis HARTMANN[a], Philip MEYER[a,b],
Florian AUER[a], Iñaki SOTO-REY[b] and Frank KRAMER[a]

[a] *IT-Infrastructure for Translational Medical Research, University of Augsburg, Germany*
[b] *Medical Data Integration Center, Institute for Digital Medicine, University Hospital Augsburg, Germany*

Abstract. Correct performance assessment is crucial for evaluating modern artificial intelligence algorithms in medicine like deep-learning based medical image segmentation models. However, there is no universal metric library in Python for standardized and reproducible evaluation. Thus, we propose our open-source publicly available Python package MISeval: a metric library for Medical Image Segmentation Evaluation. The implemented metrics can be intuitively used and easily integrated into any performance assessment pipeline. The package utilizes modern DevOps strategies to ensure functionality and stability. MISeval is available from PyPI (miseval) and GitHub: https://github.com/frankkramer-lab/miseval.

Keywords. Biomedical image segmentation; Medical Image Analysis, Reproducibility, Evaluation, Open-source framework, Performance assessment

1. Introduction

In the last decade, computer vision analysis based on artificial intelligence methods like deep learning has seen rapid growth in prediction capabilities [1]. This resulted in clinicians striving to integrate computer vision algorithms, like image segmentation, into the medical field. Medical image segmentation (MIS) covers the automated identification and annotation of medically relevant regions of interest (ROI), which can be organs, cell structures, or medical abnormalities like tumors [2]. The idea, especially in radiology and pathology, is to establish these MIS methods in their clinical routine to reduce time-consuming processes and to aid in diagnosis as well as treatment decisions [1]. However, due to the direct impact on medical decisions, the correct evaluation of MIS models is crucial. Nevertheless, recent studies indicated widespread statistical bias in evaluations of MIS models which is also caused by incorrect metric implementation [3]. Furthermore, to our knowledge, there is no universal metric library in Python for standardized and reproducible evaluation. In this work, we propose our open-source publicly available Python package MISeval, which is a metrics library for correct MIS model evaluation. It facilitates an intuitive and fast usage of various popular metrics from literature, as well as ensures implementation functionality and stability.

---

[1] Corresponding Author, Dominik Müller, IT Infrastructure for Translational Medical Research, Alter Postweg 101, 86159 Augsburg, Germany; E-mail: dominik.mueller@informatik.uni-augsburg.de

2. Methods

The open-source Python module MISeval is a metric library for **M**edical **I**mage **S**egmentation **Eval**uation. The library contains various commonly used metrics for image segmentation, which can be easily imported and instantly used for model performance assessment. MISeval is structured as an API with a central core interface for intuitive usage and is implemented in the programming language Python, which is platform-independent and highly popular for computer vision tasks. This allows simple and fast integration of MISeval in commonly used platforms like Tensorflow, PyTorch, or any NumPy-compatible image segmentation pipeline.

2.1. Metric library

Over the last decades, the MIS literature introduced a large variety of metrics for evaluation. Especially for semantic segmentation, model performance assessment can be quite complex due to the need for scoring pixel classification as well as localization correctness between predicted and annotated segmentation. The MISeval metric library contains popular metrics like Dice Similarity Coefficient (DSC), Intersection-over-Union (IoU), Sensitivity (Sens), Speecificity (Spec), Pixel Accuracy (Acc), AUC, Cohen's Kappa (Kap) and Average Hausdorff Distance (AHD), but also more complex metrics like entropy-based divergence and boundary-based distances. A summary of all metrics in MISeval, can be seen in Table 1.

2.2. Core Interface: Evaluate()

The core of our package is the *evaluate()* function, which acts as a simple and intuitive interface to access and run all implemented metrics. The desired backbone metric for the *evaluate()* function can be defined by passing the name of an already implemented metric or by passing a user-created metric function for uncomplicated integration of custom metrics. Moreover, our core function handles automatically binary as well as multi-class problems. This allows straightforward passing of any ground truth and predicted segmentation masks to the *evaluate()* function for computing the metric assessment in a single line of code.

2.3. Package Stability

Our MISeval package utilizes modern DevOps strategies to ensure package stability and functionality during ongoing development [4]. After each update, the source code is automatically built in a reproducible environment, extensively tested via unit testing, released, and, finally, deployed in the scientific community's MIS projects.

Our unit testing considers functionality, edge cases, and exceptions for each metric. For application (functionality and edge cases), multiple dummy dataset types like empty, full, or random segmentation masks as well as single and multi-class masks are tested in all combinations. For exception handling, cases with incorrect parameter usage and non-matching mask shapes are tested.

2.4. Package Availability

The MISeval package is hosted, supported, and version-controlled in the Git repository platform GitHub. This allows the utilization of platform-hosted DevOps workflows and a hub for package documentation, community contributions, bug reporting as well as feature requests. The Git repository is available under the following link: *https://github.com/frankkramer-lab/miseval.* Furthermore, MISeval is published in the Python Package Index (PyPI), which is the official third-party software repository for Python. Thus, MISeval can be directly installed and immediately used in any Python environment using "*pip install miseval*".

Our code is licensed under the open-source GNU General Public License Version 3 (GPL-3.0 License), which allows free usage and modification for anyone.

3. Results

For qualitative evaluation and functionality demonstration, we setup a deep-learning based MIS pipeline, trained a COVID-19 segmentation model for CT scans, computed predictions, and evaluated model performance using MISeval. The evaluation results are illustrated in Figure 1.The analysis utilized the MIS framework MIScnn [2] with default parameters. As dataset, we used annotated computed tomography scans of COVID-19 positive patients from Ma et al. [5].

For quantitative evaluation, we compared our metric library with other widely used frameworks for machine learning and image analysis. As it can be seen in Table 1,
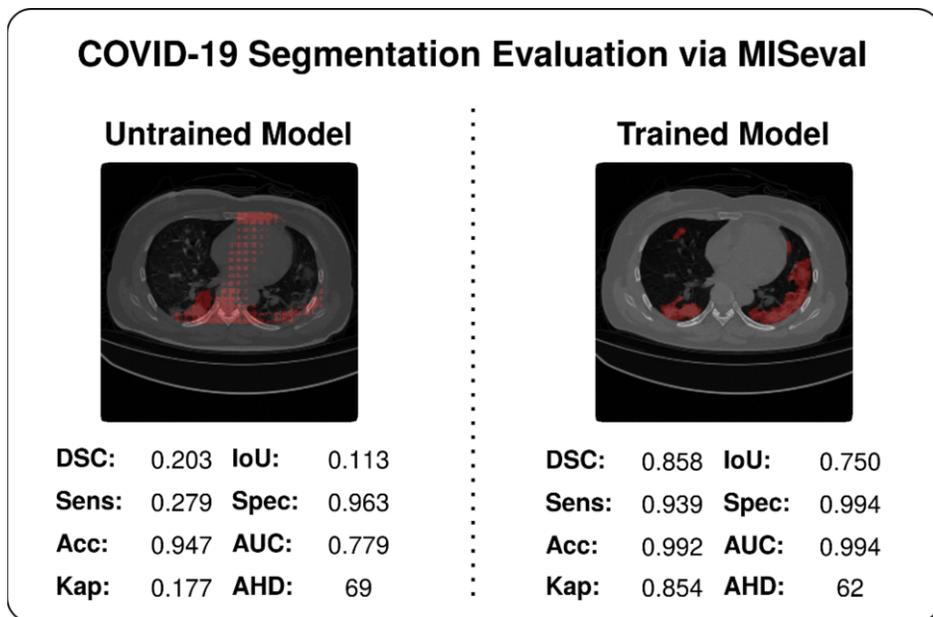

Figure 1. Illustration of various selected metrics from the library of MISeval to evaluate model performance on the use case COVID-19 infected region segmentation. The figure compares an untrained model (after 1 epoch during training) to a fully trained model (after 163 epochs) and shows computed tomography scans for each model with predicted infected regions (red).

MISeval provides currently 28 metrics, which is the highest number of segmentation metrics compared to other analyzed frameworks: scikit-learn [6] with 18, EvaluateSegmentation from VISCERAL [7] with 13, PyMIA [8] with 12, Tensorflow [9] with 16 and TorchMetrics [10] with 12.

Table 1. Overview and comparison of currently implemented metrics in MISeval. For in-detail formula description and theory of the majority of presented metrics, we refer to the review from Taha et al. [4].

Group	Metric	scikit-learn	VISCERAL	PyMIA	Tensorflow	TorchMetrics	MISeval
Spatial Overlap	Dice Similarity Coefficient / F1-score	X	X	X	X	X	X
	Intersection-Over-Union / Jaccard Index	X	X	X	X	X	X
	Sensitivity / Recall	X	X	X	X	X	X
	Specificity		X	X	X	X	X
	Precision	X	X	X	X	X	X
Spatial Distance	(Average) Hausdorff		X	X			X
	Bhattacharyya						X
	Canberra						X
	Chebyshev						X
	Chi Square	X					X
	Cosine	X			X		X
	Euclidean	X					X
	Manhattan	X			X		X
	Hamming	X			X	X	X
	Mahanabolis		X				
	Minkowski						X
	MAE / MSE	X		X	X		X
	Pearson						X
Correlation	Interclass Correlation		X	X			
	Matthews Correlation	X			X	X	X
Divergence	Jensen-Shannon						X
	Kullback-Leibler				X	X	X
	Cross-Entropy	X			X		X
	Hinge	X			X	X	X
Probabilistic or Pairing	AUC	X	X	X	X	X	X
	Cohen Kappa	X	X	X	X	X	X
	Accuracy / Rand Index	X	X	X	X	X	X
	Balanced Accuracy	X					X
	Adjusted Rand Index	X	X	X			X
Volume	Volumetric Similarity		X				X

4. Discussion

Our proposed package MISeval allows a universal, reproducible, and standardized application of various metrics for MIS evaluation, which hopefully reduces the risk of statistical bias in studies through incorrect custom implementations. By following the state-of-the-art package stability and availability strategies, MISeval has the potential to be integrated into any future scientific performance analysis due to package stability, easy accessibility, and further contribution possibilities.

Our road map and future direction for MISeval is to ensure ongoing support, the further extension of our metric library, and providing guidelines on correct metric usage as well as evaluation. Furthermore, we plan to propose a new metric similar to the Dice Similarity Coefficient for handling the current issue of evaluating non-present classes in ground truth annotations like in control samples.

5. Conclusions

In this work, we proposed our open-source Python package MISeval: a metric library for medical image segmentation evaluation. The library contains various popular metrics which can be easily used and integrated into any performance assessment for image segmentation models. MISeval can be directly installed as a Python library from PyPI (miseval) and is available in GitHub: *https://github.com/frankkramer-lab/miseval*.

References

[1]   Litjens G, Kooi T, Bejnordi BE, Arindra A, Setio A, Ciompi F, et al. A survey on deep learning in medical image analysis. 2017;42(December 2017):60–88.
[2]   Müller D, Kramer F. MIScnn : a framework for medical image segmentation with convolutional neural networks and deep learning. BMC Med Imaging. 2021 Jan 21;21:12.
[3]   Zhang Y, Mehta S, Caspi A. Rethinking Semantic Segmentation Evaluation for Explainability and Model Selection.
[4]   Dyck A, Penners R, Lichter H. Towards definitions for release engineering and DevOps. In: Proceedings - 3rd International Workshop on Release Engineering, RELENG 2015. Institute of Electrical and Electronics Engineers Inc.; 2015. p. 3.
[5]   Ma J, Ge C, Wang Y, An X, Gao J, Yu Z, et al. COVID-19 CT Lung and Infection Segmentation Dataset [Internet]. Zenodo; 2020. Available from: https://doi.org/10.5281/zenodo.3757476
[6]   Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in {P}ython. J Mach Learn Res. 2011;12:2825–30.
[7]   Taha AA, Hanbury A. Metrics for evaluating 3D medical image segmentation: Analysis, selection, and tool. BMC Med Imaging [Internet]. 2015 Aug 12 [cited 2021 May 14];15(1):29. Available from: http://bmcmedimaging.biomedcentral.com/articles/10.1186/s12880-015-0068-x
[8]   Jungo A, Scheidegger O, Reyes M, Balsiger F. pymia: A Python package for data handling and evaluation in deep learning-based medical image analysis. Comput Methods Programs Biomed. 2021 Jan 1;198:105796.
[9]   Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems [Internet]. 2015. Available from: https://www.tensorflow.org/
[10]   Detlefsen N, Borovec J, Schock J, Jha A, Koker T, Di Liello L, et al. TorchMetrics - Measuring Reproducibility in PyTorch. J Open Source Softw [Internet]. 2022 Feb 11 [cited 2022 Mar 19];7(70):4101. Available from: http://arxiv.org/abs/1907.11692