DebiAI: Open-Source Toolkit for Data Analysis, Visualisation and Evaluation in Machine Learning

Tom Mansion
IRT SystemX
91120 Palaiseau, France
tom.mansion@irt-systemx.fr

Raphaël Braud
IRT SystemX
91120 Palaiseau, France
raphael.braud@irt-systemx.fr

Ahmed Amrani
IRT SystemX
91120 Palaiseau, France
ahmed.amrani@irt-systemx.fr

Sabrina Chaouche
IRT SystemX
91120 Palaiseau, France
sabrina.chaouche@irt-systemx.fr

Faouzi Adjed
IRT SystemX
91120 Palaiseau, France
faouzi.adjed@irt-systemx.fr

Loïc Cantat
IRT SystemX
91120 Palaiseau, France
loic.cantat@irt-systemx.fr

***Abstract*—DebiAI is an open-source tool designed for data analysis, visualization, as well as evaluation and comparison of Machine Learning (ML) models. It is intended to be used both at the stage of the project data preparation, and for the evaluation of the ML models performances. It has a rich and user-friendly graphical interface that allows to visualize, analyze, select, edit and annotate data and metadata, as well as for bias detection and contextual evaluation of ML models. The tool relies on a generic data model, making it applicable to any type of ML task: classification, regression, object detection in images and more. It is an open source code distributed under the Apache License, Version 2.0 . The code is publicly available on https://github.com/debiai and further information along with guidelines for the users can be found on its dedicated website https://debiai.irt-systemx.fr.**

Index Terms*—*Data Analysis; Data Visualization; Bias Detection; Contextual Evaluation; Machine Learning; Trustworthy AI.

I. Introduction

Data analysis and visualization are fundamental tasks in Machine Learning (ML) projects. They are playing a crucial role in a typical ML process, and are contributing not only in the data preparation phase but also during and after the model building. This emerging research topic, which combines several interactive systems and domains for ML processes, focused on human interaction and collaboration [1] created a new field that is Human-Centred Machine Learning (HCML) interaction [2]. Thus, a typical HCML framework allows an interactive visual analytic and ML evaluation [3]. Consequently, efficient tools are needed to assist, during the whole ML process, in data preparation and quality inspection before and after using the data for model training.

An effective tool should help enhance the iterative ML process across various phases: data preparation, analysis, anomaly detection, and annotation during the preparation phase, as well as the analysis and evaluation of models results. This aids in identifying the model's weaknesses and detecting issues at the data level. In real-world ML projects (e.g., industrial ones), the data is enriched by metadata including operational context, expert knowledge, etc; that can help better understand the raw data and can be informative for the learning process, and hence improving the quality of the model's learning and predictions. Having such tools contribute also to the enhancing the trustworthiness of the used ML algorithm.

In ML-based engineering systems, it is crucial to guarantee key properties like accuracy, robustness, explainability, fairness, privacy, among many other primary values of Trustworthiness. Current research and development challenges of deploying trustworthy ML solutions are covered by wide programs like Confiance.ai [4], the French AI flagship program to industrialize trustworthy AI-based critical systems [5] [6] and the TAILOR [7] network at the European level.

DebiAI has been developed by the IRT SystemX in the framework of Confiance.ai program to contribute in ensuring trustworthiness by data, and serves as the main interface to view, analyze, select, edit and/or annotate any type of data and metadata. The rest of the paper is organized as following: Section II presents a brief state of the art of developed HCML tools, Section III presents methodology, followed by implementation description and application in a real use case described in Sections IV and V, respectively. Finally, conclusions and perspectives are drawn in Section VI.

II. Literature review

Data visualization is the practice of representing information using graphical representations, employing technology-driven tools and software. Its fundamental objective is to enhance pattern recognition, improve understanding of complex concepts and facilitate in-depth exploration, thereby fostering the generation of new insights. Well-designed data visualizations can help in understanding large datasets and establishing connections between ideas, concepts, and processing stages. Therefore, visual analysis can contribute to the optimization of AI approaches by actively participating in all aspects of the model building process [8] [9]. Similarly, Hohman et al. [10] highlights that successful ML applications often require iterations in data handling and continuous adjustments of the model. The authors introduced CHAMELEON, an interactive tool designed to attribute data iteration, thereby enhancing model performance, data validation, and the overall quality of ML projects. To improve data quality, Kandel et al. [11] presented Profiler, a tool using data mining to automatically detect issues and recommend coordinated visualizations for context-based assessment. Profiler offers methods for integrated statistical and visual analysis and view suggestions. ScrutinAI [12] is a Visual Analytics tool specifically tailored for enhancing the comprehension of deep neural network (DNN) predictions. Its primary objective is to identify and investigate potential weaknesses within models. To facilitate this, ScrutinAI provides interactive visualizations of input and output data, along with interactive plots and data filtering for comprehensive analysis of predictions. This tool is specifically designed for object detection and semantic segmentation, whereas DebiAI is applicable to a wide range of use cases. Zhang et al. [13] presented Manifold, a visual analytics platform designed for comparing and debugging ML models. The platform empowers users to categorize instances based on the model's accuracy and confidence, identify symptomatic instances that generate incorrect results and continually help to enhance the model's performance. As DebiAI, Manifold is created as a generic tool that operates independently of the internal logic of the ML model. It focuses on the input and the output. In order to improve the models' performances and gain insights into their limitations, relying solely on the overall result from the test and training sets is insufficient. To address this limitation, the ModelTracker [14] tool offers instance-level visualization of results and enables users to individually inspect each instance. The tool has been applied to the binary classification task. In [15], the authors applied the approach for a multi-class classification task. They used Parallel Coordinates Plots (PCP) to visually represent the multi-class predictions for a subset of instances. To complete these shortcomings and improve the understanding of models results, the proposed DebiAI enables the analysis of model's outcomes across various levels of granularity (instance, subset and dataset). This functionality has been applied to multiple tasks such as regression, classification, object detection, etc. In line with the approach outlined in [15], DebiAI leverages the use of PCP for model's results analysis. However, the implementation of PCP within DebiAI is flexible, enabling its use not only for result analysis but also for assessing attributes.

III. Methodology

DebiAI is a web-based visual analytics application designed to support ML and data analysis. Its emphasis lies in two crucial phases of the ML pipeline: pre-model and post-model building. It facilitates the development of ML models by assisting in data analysis during data curating and processing stage and enabling models performances comparison.

In the pre-model construction phase, DebiAI serves as a key resource for data scientists and ML engineers during project preparation. It enables them to visually identify biases and errors in data inputs, detect anomalies and outliers throughout the data life cycle, assess data quality and domain coverage through relevant metrics and select and analyze subsets of data to improve the quality of ML models.

In the post-model building phase, DebiAI serves as a visual analytics solution, simplifying the interpretation of the ML model's outputs. Its primary objective is to present the model's results in an intuitive and easily understandable manner, ultimately enhancing user confidence in the model's predictions. Additionally, DebiAI offers features to identify model's weaknesses, comparing performances, and evaluating model's effectiveness. This comprehensive approach fosters ongoing model refinement, tailored to the specific needs of the use case.

In both phases, DebiAI provides tools for creating and sharing statistical visualizations of the project data and results with collaborators (team or/and clients).

A. Functional Description

DebiAI is an intuitive visualization tool designed to simplify the creation of interactive dashboards, empowering users without little to no programming skills. It offers a diverse set of graphical widgets, including charts, tables, parallel categories, parallel coordinate plots, interval plots, night stars plots, and sample arrays. Moreover, DebiAI provides a user-friendly and flexible solution for interactive dashboard design, allowing users to effortlessly configure, adjust, resize, and position these widgets within their dashboards, ensuring the utmost customization of data presentation. This includes the ability to generate and share statistical visualizations of project data with team members or clients, fostering collaboration and informed decision-making by providing clear insights into the data. One of DebiAI's standout features is its dynamic data selection and filtering capabilities, encouraging continuous exploration. Users can effortlessly create data subsets (selections) and apply filters based on various variables and contexts. This ensures that the dashboard consistently presents the identified subset of data. Furthermore, DebiAI assists users in identifying biases and inaccuracies in inputs, results, project data contexts or ground truths, thereby improving data integrity. DebiAI facilitates the evaluation and comparison of ML model's performances within the whole dataset or a specific data subset. It enables the analysis of results across multiple levels of granularity. Indeed, the model's performances are calculated at the level of each instance. Consequently, it is possible to identify the contexts or a combination of contexts in which the model encounters difficulties. It also simplifies the generation and organization of datasets, supporting in-depth analysis and potential retraining. DebiAI relies on a generic data model that facilitates seamless application across various datasets, data types and use cases while maintaining consistent data processing practices. This essential feature provides DebiAI with flexibility, allowing it to transition effortlessly between various datasets or model's results. In addition to its visualization capabilities, DebiAI incorporates implementations of statistical measures such as correlation analysis using Pearson or Spearman coefficients. To support these visualizations, DebiAI also integrates techniques for discretizing continuous variables. In addition, it enables the use of internal or external algorithms to compute metrics or indicators on the data. Consequently, these metrics can be calculated either before integrating the data into DebiAI or during the data analysis phase. Various types of calculations can potentially be carried out by these algorithms, including the computation of new features, the assessment of model's results quality, as well as indicators of data quality and distribution.

IV. Implementation Description

In this section, we describe DebiAI's global architecture and further dive into the detail of each component.

A. DebiAI Technical architecture

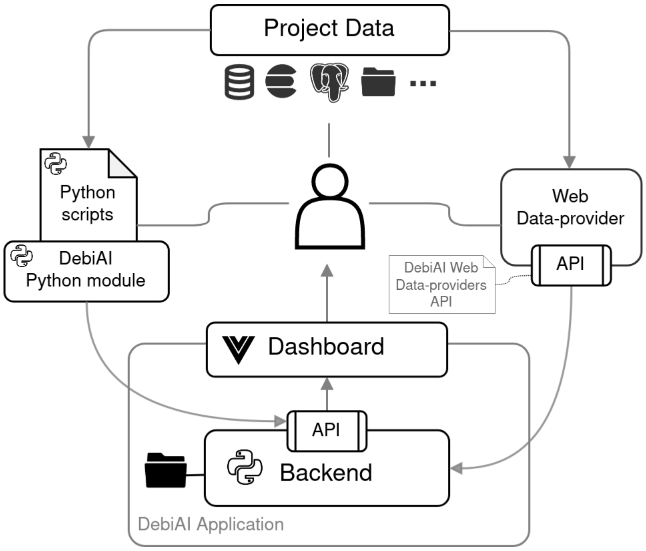


Fig. 1: DebiAI architecture overview

The architecture of DebiAI is bifurcated into two main environments (Fig. 1):

The project environment :

- Project Data: This is the source of data that the user intends to analyze. It may originate from a variety of sources and formats, such as CSV or JSON.
- Python Scripts and DebiAI Python Module: Using the DebiAI Python module, users can adapt their existing scripts and workflows to create selections and insert data and model's results into DebiAI.
- Web Data-Provider: These are the services created by the user's project that enable DebiAI to fetch data directly from the project's data sources. A Web Data-Provider can be developed using any programming language, access data from any type of database, and be hosted on any server. The only stipulation is that it should implement and expose a specific REST API according to a defined contract.

DebiAI application environment:

- DebiAI Web Dashboard: This is the user interface of DebiAI, developed using VueJs. It provides users with an interactive platform to manage and view their data, and is hosted and served by the DebiAI backend.
- Backend and API: This is a Python-powered backend that not only provides an API but also serves the Web dashboard. This API is employed by the dashboard for data retrieval and by the Python module for data insertion. Additionally, it manages communications with the Web Data-providers and processes computational requests made by the dashboard.
- Data storage: DebiAI uses a folder-based data store that contains data in a JSON format. This data store supports the DebiAI backend by retaining projects created by the Python module and some specific dashboard elements, including layout configurations for project dashboards.

B. DebiAI Generic Data Model

The data format required must follow a CSV structure with no missing values. One of the most important features of DebiAI is its data model. The main objective is to enable the determination of the format of instances and the relationship between instances, models, models' outputs, and models' evaluation metrics per instance. Syntactically, each instance is composed of attributes, contexts and annotations. The instance is linked to multiple ML models, where each model produces an output. Evaluation metrics are also associated with the model's outputs. This structure is applicable to all types of data and ML tasks (classification, regression, object detection, etc.).

C. DebiAI Data Integration Process

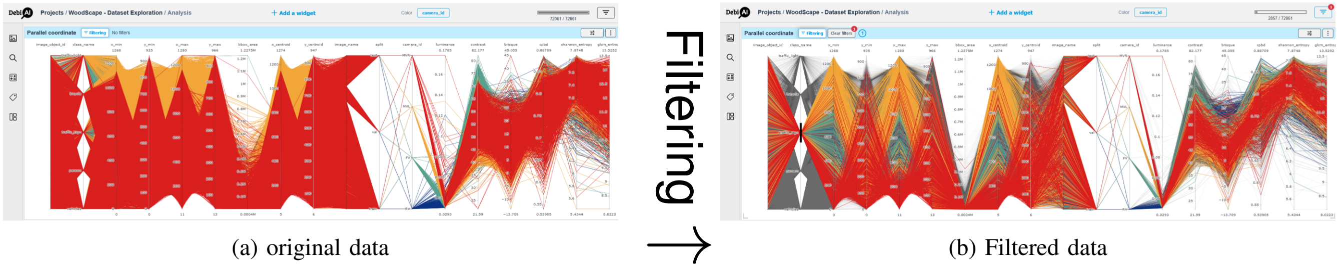
DebiAI offers two main ways to add data, each suited for different types of users and projects:

1) **Python Module:** This principal method enables seamless integration of project's data into DebiAI via a dedicated Python module. Made for an integration within Python workflows, this approach, for example, facilitates the direct transfer of models' results post-evaluation. This method is especially handy for those who primarily use Python.
2) **Data Providers:** Alternatively, DebiAI can interface with data through RESTful services, termed 'Data Providers'. This method is database-agnostic, allowing DebiAI to directly request project's data, thereby making the data loading process faster and more efficient. Unlike the Python module, it doesn't require DebiAI to duplicate data within its integrated database. Although setting up a Data Provider is more time-consuming than using the Python module, it offers greater efficiency and flexibility. This is particularly beneficial for long-term projects that regularly update their data.

Each method offers distinct benefits, and the choice depends on the specific requirements and scale of the project.

V. DebiAI Application

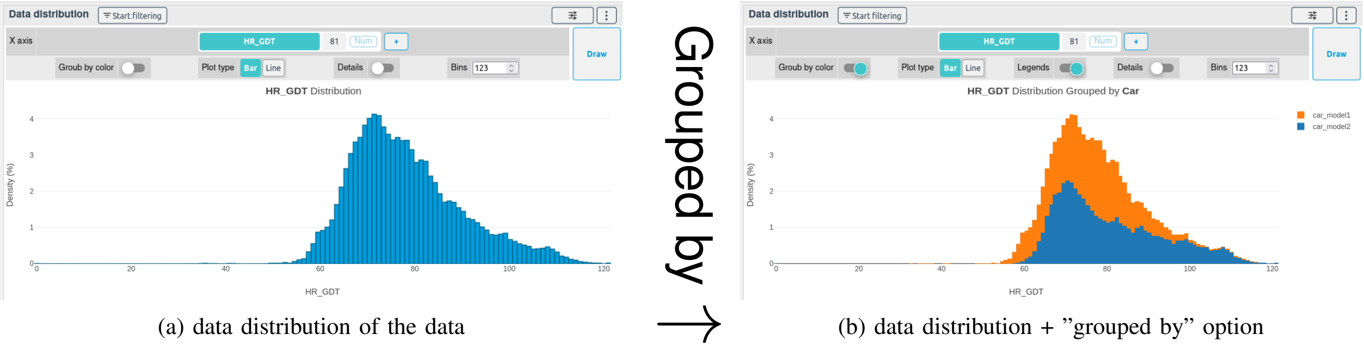
DebiAI is built upon a generic data model, and does not depend on data type (for instance images and time series),



(a) original data

(b) Filtered data

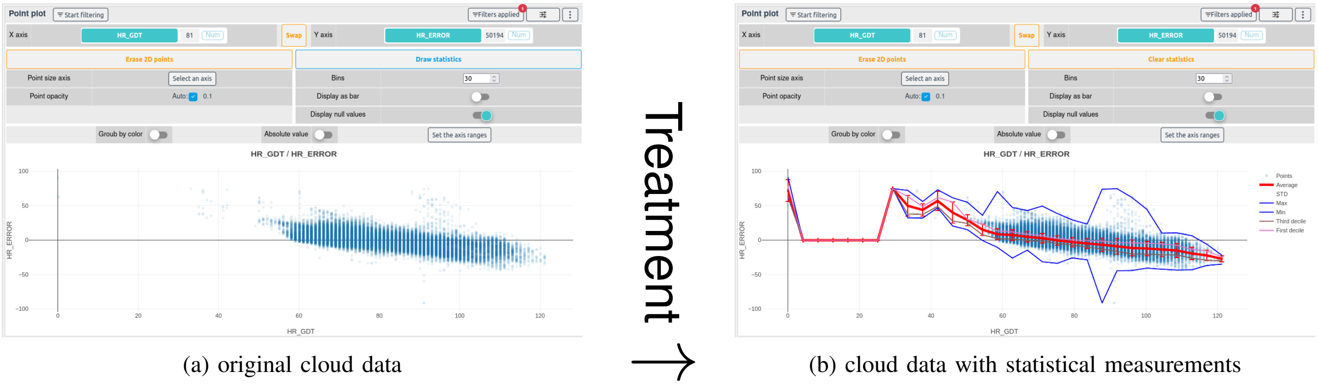
Fig. 2: parallel coordinates widget. (a) represents the original data uploaded and (b) represents the same widget by selecting a subset of variables interactively.



(a) data distribution of the data

(b) data distribution + ”grouped by” option

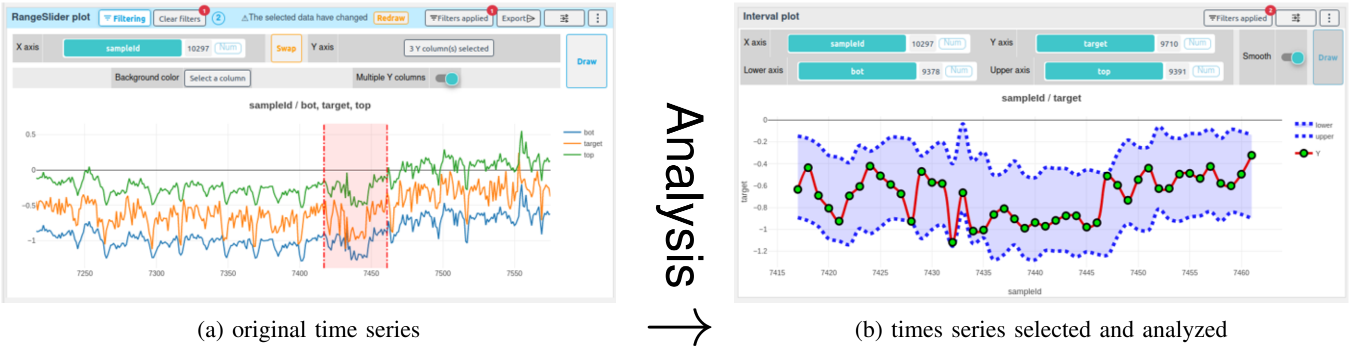
Fig. 3: illustration of data distribution by adding the option of ”grouped by”. (a) represents an example of data and (b) represents the same data grouped by another variable with two different colors.



(a) original cloud data

(b) cloud data with statistical measurements

Fig. 4: Statistical treatment for an example of cloud data. (a) represents the original data and (b) represents the data by adding a set of statistical measures. Here illustrated measures are: mean, standard deviation, min and max and deciles



(a) original time series

(b) times series selected and analyzed

Fig. 5: Statistical analysis applied on times series example. (a) represents the original time series (b) represents the times series filtered and analyzed by adding a set of statistical measures

making it pertinent to various use cases across a multitude of datasets. This intrinsic adaptability allows it to be valuable in a wide range of scenarios. It demonstrates its utility in the analysis of time series data, simplifying essential tasks such as regression. Furthermore, its functionality seamlessly extends to computer vision applications. Indeed, DebiAI provides tailored visual support for each stage of the process, enhancing models in tasks such as object detection and image classification.

In the following two sections, we will present the use of multiple widgets in DebiAI for various use cases and provide an overview of a use case related to 2D object detection.

A. *DebiAI Visual Functionalities*

As described in Section III-A, DebiAI gives the ability to visualize and create interactive dashboards. Moreover, it can visualize various data types such as time series, point clouds and tabular data and display computed attributes of images. However, for images viewing, it can establish links with external tools. In this section, we review a set of graphics implemented on different datasets with different data types. We also illustrate the main filtering features proposed by DebiAI. Four graphical visualizations are presented by exploring the parallel coordinates, the data distribution, the points plots and the time series widgets enhanced with interactive options. Fig. 2 illustrates a visualization of a dataset by using a parallel coordinates and the possibility to filter directly a set of variables. Another graphic visualization to analyse data distributions variables with the possibility of grouping by other variables is shown in Fig. 3. The third visualization selected from DebiAI is the possibility to apply statistical measures. Fig. 4, captures a data cloud visualized with its primary statistical measures; an envelope of min and max of the data, the average, the confidence interval $\pm\sigma$, where σ represents the standard deviation and also two deciles of the data. Fig.5 visually encapsulates the two distinct stages in the statistical analysis of a time series. Initially, Fig.5a displays the original data over an extended period. Subsequently, in the second stage, Fig.5b illustrates the time series after a more detailed analysis and filtering, focusing on a shorter timeframe. Among the statistical measures, a noteworthy one to explore when analyzing dataset's variables is examining their correlation, a task effortlessly accomplished using the DebiAI's correlation matrix widget.

B. *WoodScape Dataset: 2D Object Detection Case Study*

The WoodScape dataset [16] is a public dataset containing more than 100.000 images of urban scenes captured using fish-eye cameras for automotive driving tasks. The images are provided by 4 different cameras with different angles of view (front, rear, middle right and middle left) with 360° coverage and have annotations for a diverse set of computer vision tasks.

In our study, we focus on the 2D bounding boxes detection task with five classes: vehicle, person, bicycle, traffic light and traffic sign. We applied two versions of YOLO-based architectures, specifically YOLOv5 and YOLOv8. A YOLOv5 and a YOLOv8 models trained on WoodScape dataset and a YOLOv8 model pre-trained on COCO2017 [17].

The first step is to obtain a comprehensive overview of the data distribution, understand its scope and how it can be effectively used in a ML process. This comprehension is crucial, as it helps in the formulation of effective training strategies. The Fig. 6 shows the distribution of the dataset's train set composition of each class of the five cited above grouped by the camera id. By applying the same configuration to display the distribution of each of the three sets (train, validation and test), we observe a similar distribution among the three of them. This confirms the appropriate split of the data. Nevertheless, the figure highlights the imbalance among the distributions of the five classes, suggesting the need for adaptive training techniques, for example using a weight sampler. It is essential to consider this imbalance when interpreting the models outputs to avoid biased and skewed conclusions.

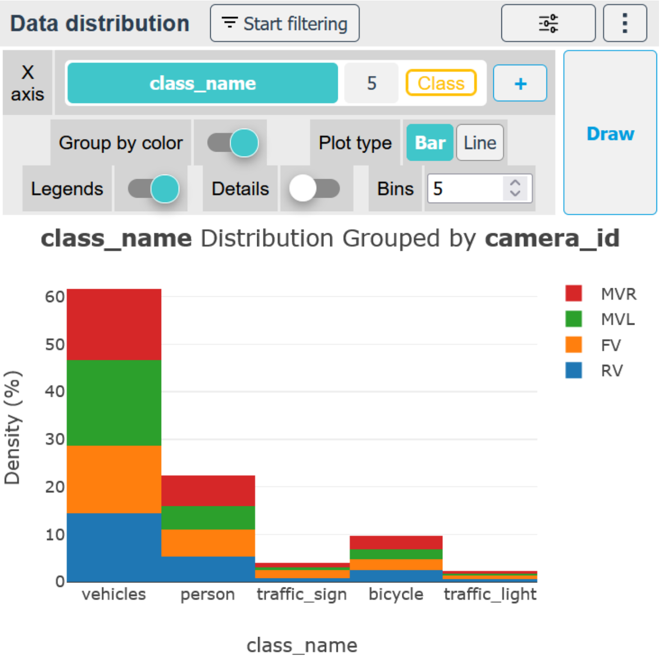

Data distribution
X axis: class_name 5 Class
Group by color: on
Plot type: Bar
Legends: on
Details: off
Bins: 5

Fig. 6: Dataset's train set split by objects class distribution grouped by Camera ID

In the second stage, we used DebiAI to analyze the results of our models applied on the WoodScape test set and put them back into the context of the dataset and its features. This approach ensures an accurate interpretation of the models' outputs and provides potential improvements directions. The Fig. 7 shows the distribution of the f1-score of each model grouped by the camera id, where we can easily spot the gap in performance between the three models: having the two models trained on WoodScape dataset showing higher scores compared to the one pre-trained on COCO 2017 dataset, which is expected giving the discrepancy between the two datasets. We can also notice that the Yolov5 trained on the WoodScape

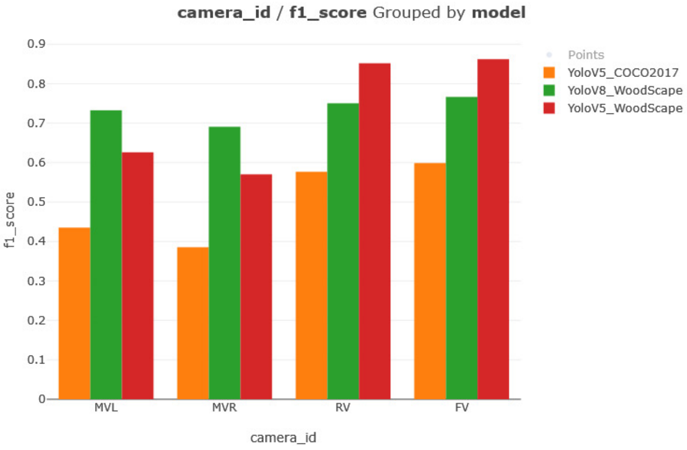


Fig. 7: f1-score results by Camera ID grouped by Models

train set has better score on the data coming from the front and rear view cameras (FV & RV) of the vehicle while the Yolov8 also trained on WoodScape shows a greater score on the middle view cameras (left & right) data. This first observations led to further investigations using DebiAI in an attempt to understand the models' outputs; you can check our demo on this case study on our website for more details.

VI. Conclusions and Perspectives

In this paper, we introduced DebiAI, a versatile web-based visual analytics tool that enhances data preparation process, quality assessment, model results analysis and comparison in ML projects. Its adaptability to various use cases and user-friendliness make it a valuable asset contributing to the trustworthiness in AI. For instance, we illustrated its application in a use case of 2D object detection task for driving assistance. As Machine Learning evolves, DebiAI can play a pivotal role in ensuring reliable and interpretable ML outcomes, solidifying its relevance in the field. In DebiAI's outlook, the priorities are to simplify the data integration process, enhance interoperability with the learning process to retrieve and analyze data from each cycle. The concepts of robustness and explainability are also tied to model's quality. Therefore, incorporating these metrics into the process is critical for overall trust. Furthermore, we are considering coupling DebiAI with a data version control system to ensure traceability of dataset and models' results evolution.

Acknowledgment

This work has been supported by the French Government under the "France 2030" program, as part of the SystemX Technological Research Institute. This work was conducted as part of the Confiance.ai program, which aims to develop innovative solutions for enhancing the reliability and trustworthiness of industrial AI-based systems.

References

[1] F. Sperrle, M. El-Assady, G. Guo, R. Borgo, D. H. Chau, A. Endert, and D. Keim, "A Survey of Human-Centered Evaluations in Human-Centered Machine Learning," *Computer Graphics Forum*, vol. 40, no. 3, pp. 543–568, Jun. 2021, publisher: John Wiley & Sons, Ltd. [Online]. Available: https://doi.org/10.1111/cgf.14329

[2] T. Kaluarachchi, A. Reis, and S. Nanayakkara, "A review of recent deep learning approaches in human-centered machine learning," *Sensors*, vol. 21, no. 7, p. 2514, 2021.

[3] D. Sacha, M. Sedlmair, L. Zhang, J. A. Lee, J. Peltonen, D. Weiskopf, S. C. North, and D. A. Keim, "What you see is what you can change: Human-centered machine learning by interactive visualization," *Neurocomputing*, vol. 268, pp. 164–175, 2017.

[4] Confiance.ai, "Confiance.ai Program," https:www.confiance.ai, 2024, online; accessed 19-February-2024.

[5] M. Adedjouma, C. Alix, L. Cantat, E. Jenn, J. Mattioli, B. Robert, F. Tschirhart, and J.-L. Voirin, "Engineering Dependable AI Systems," in *17th Annual System of Systems Engineering Conference (SOSE)*. Rochester, United States: IEEE, Jun. 2022. [Online]. Available: https://hal.science/hal-03700300

[6] M. Juliette, L. R. Xavier, B. Bertrand, C. Loic, T. Fabien, R. Boris, G. Rodolphe, and N. Yves, "Ai engineering to deploy reliable ai in industry," *AI4I*, vol. 25, no. 1, 2023.

[7] Taylor, "TAYLOR Network Program," https://tailor-network.eu/, 2024, online; accessed 19-February-2024.

[8] J. Yuan, C. Chen, W. Yang, M. Liu, J. Xia, and S. Liu, "A survey of visual analytics techniques for machine learning," *Computational Visual Media*, vol. 7, no. 1, pp. 3–36, Mar. 2021. [Online]. Available: https://doi.org/10.1007/s41095-020-0191-7

[9] J. Wang, S. Liu, and W. Zhang, "Visual analytics for machine learning: A data perspective survey," *CoRR*, vol. abs/2307.07712, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2307.07712

[10] F. Hohman, K. Wongsuphasawat, M. B. Kery, and K. Patel, "Understanding and visualizing data iteration in machine learning," in *Proceedings of the 2020 CHI conference on human factors in computing systems*, 2020, pp. 1–13.

[11] S. Kandel, R. Parikh, A. Paepcke, J. M. Hellerstein, and J. Heer, "Profiler: Integrated statistical analysis and visualization for data quality assessment," in *Proceedings of the International Working Conference on Advanced Visual Interfaces*, 2012, pp. 547–554.

[12] E. Haedecke, M. Mock, and M. Akila, "ScrutinAI: A visual analytics tool supporting semantic assessments of object detection models," *Computers & Graphics*, vol. 114, pp. 265–275, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S009784932300105X

[13] J. Zhang, Y. Wang, P. Molino, L. Li, and D. S. Ebert, "Manifold: A model-agnostic framework for interpretation and diagnosis of machine learning models," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 364–373, 2019.

[14] S. Amershi, M. Chickering, S. Drucker, B. Lee, P. Simard, and J. Suh, "ModelTracker: Redesigning Performance Analysis Tools for Machine Learning," in *Proceedings of the Conference on Human Factors in Computing Systems (CHI 2015)*. ACM - Association for Computing Machinery, Apr. 2015, edition: Proceedings of the Conference on Human Factors in Computing Systems (CHI 2015). [Online]. Available: https://www.microsoft.com/en-us/research/publication/modeltracker-redesigning-performance-analysis-tools-for-machine-learning/

[15] D. Ren, S. Amershi, B. Lee, J. Suh, and J. D. Williams, "Squares: Supporting interactive performance analysis for multiclass classifiers," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 01, pp. 61–70, jan 2017.

[16] S. Yogamani, C. Hughes, J. Horgan, G. Sistu, P. Varley, D. O'Dea, M. Uricár, S. Milz, M. Simon, K. Amende *et al.*, "Woodscape: A multi-task, multi-camera fisheye dataset for autonomous driving," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9308–9318.

[17] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft coco: Common objects in context," 2015.