Platform to help CNN training in manufacturing teaching environments

Thiago Rodrigo Monteiro Salgado
CETELI-UFAM
Federal University of Amazonas
Manaus – AM, Brazil
0009-0000-3456-8452

Mattheus Smith da Costa Fernandes
UFAM
Federal University of Amazonas
Manaus – AM, Brazil
mscf@icomp.ufam.edu.br

Guido Soprano Machado
CETELI-PPGEE-UFAM
Federal University of Amazonas
Manaus – AM, Brazil
0000-0002-2365-9735

Vicente Ferreira de Lucena Jr
CETELI-PPGEE-UFAM
Federal University of Amazonas
Manaus – AM, Brazil
0000-0001-9864-2850

***Abstract* — This research-to-practice full paper presents a platform developed to support AI teaching, training, and evaluation. Artificial intelligence (AI) has revolutionized several fields, including pattern recognition, natural language processing, and computer vision. In this context, convolutional neural networks (CNNs) have been effectively utilized to detect objects in images. In industry, AI has the potential to optimize processes and increase efficiency, becoming an essential factor in the evolution of manufacturing systems. However, the training of AI models is complex and requires a sophisticated level of knowledge for data manipulation and an understanding of model training. To simplify this process, a web-based platform has been proposed to facilitate the training of YOLO networks. The platform was developed via the Streamlit library, which provides an interface to guide users through model training. The platform aims to lead users through six steps to obtain and deploy a model directly through the platform, enabling performance evaluation and metric tracking.**

***Keywords* — Education platform, AI in Education, Engineering Teaching, CNN.**

I. Introduction

Artificial intelligence (AI) is a fundamental tool in various fields, generating significant advancements in pattern recognition, natural language processing, and computer vision [1, 2]. Among AI techniques, convolutional neural networks (CNNs) have excelled in computer vision because of their ability to learn complex image patterns [3].

In the industrial context, the application of AI has the potential to optimize processes, increase efficiency, and reduce costs [4, 5]. However, training AI models, such as CNNs, can be challenging, requiring technical expertise and prior knowledge of data organization and manipulation [6, 7]. To make this process more educational, accessible, and practical, this paper proposes a platform that simplifies the training of You Only Look Once (YOLO) networks, a famous architecture for object detection in images, which delivers highly satisfactory and rapid results [8].

We want to investigate how to train engineers in manufacturing automation efficiently, an open problem demanding constant research [9, 10]. In fact, one of the proposals of this paper to facilitate training and understanding of the process involves executing six main steps that are common during AI development. This is achieved through the integration of Python libraries, such as Ultralytics, for training and configuration and the use of external tools for data labeling (in our case, images), such as Makesense. These tools enable users to label images efficiently and accurately without requiring advanced image processing or programming knowledge. This integration and step-by-step guidance significantly contribute to the learning process.

This work proposes an accessible platform for training object detection models in images, aiming to provide a solution to help industry professionals develop and implement automated inspection systems efficiently and cost-effectively through networks trained by the platform. To demonstrate the effectiveness and relevance of the proposed system, experiments were conducted using didactic plants to obtain a representative dataset. The results indicated the system's ability to train AI models with high performance in object detection during test inference.

This paper is organized into two sections. Section II presents the background and provides the necessary concepts for understanding the platform. Section III describes the proposed system. Section IV details the development of the platform and its functionality. Section V presents the results obtained in this work and provides conclusions.

II. Background

This chapter presents the fundamental technologies used in developing the proposed platform, highlighting their functionalities and roles within the system.

A. Streamlit: Interactive and Intuitive Frontend

The main advantage of Streamlit lies in its ability to transform Python scripts into interactive web applications in minutes [11]. Its simple and intuitive syntax allows developers to focus on the application logic, whereas Streamlit automatically manages the rendering and updating of the user interface.

Streamlit also provides built-in support for running applications on a local network, making it easy to access and share the application within an organization. By executing a command, users can start the Streamlit server and access the application via the host machine's IP address, ensuring accessibility to all collaborators on the same network.

B. YOLO (You Only Look Once) and the Ultralytics library

YOLO is a convolutional neural network architecture widely used for real-time object detection in images. Its unique approach divides the image into a grid and simultaneously predicts bounding boxes and class probabilities for each grid cell, allowing fast and accurate object detection in images [12].

Implementing artificial intelligence is not a trivial task. To simplify the implementation of YOLO, both for students and for use in the system, the Ultralytics Python library was utilized. This library provides an easy-to-use interface for training, testing, and using YOLO models, offering ease of use and support from the Ultralytics repository [13]. It provides pretrained network implementations of various YOLO versions and tools for custom training with user-provided datasets.

By integrating YOLO and the Ultralytics library into our system, we streamlined the training of new object detection models and implemented real-time detection in our applications. This contributes to the efficiency and accuracy of our computer vision solutions.

C. SQL Database: Backend for Pretrained Network Management

For the storage and management of pretrained networks available on the device and the results of their inferences, we utilize an SQL database. Structured Query Language (SQL) is a standardized query language used to interact with relational databases.

The SQL database is a fundamental system component, enabling secure and organized storage of pretrained networks and related information such as metadata and performance metrics. Additionally, the use of an SQL database provides flexibility in data querying and retrieval, facilitating system integration and expansion in the future.

One of the critical features of our system is its ability to register a new network after training automatically. This is achievable through the integration between the training process and the SQL database, where the data from the trained model are automatically recorded and stored for future reference.

III. Proposed System

A. Typical Steps in Training an AI Model

Training an artificial intelligence (AI) model consists of three steps: data preparation, model training, and model evaluation [14, 15]. Each of these steps, as depicted in Fig. 1, is particularly important in developing an accurate and effective AI model.

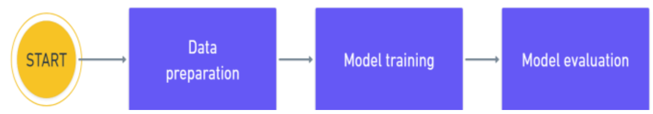

Fig. 1. Steps of AI training

1) ***Data Preparation:*** Data preparation is the first step in the training process for an AI. In this phase, relevant data for the problem are collected, organized, and preprocessed to ensure its quality and usefulness in model training. This may include data cleaning, normalization, feature selection, and splitting the data into training, validation, and testing sets. In this work, we performed only data splitting [16].

2) ***Model training:*** the next step is training the AI model after data preparation. In this stage, the model is fed with the training data and iteratively adjusted to learn patterns and relationships within the data. This is accomplished through machine learning algorithms, which optimize the model parameters on the basis of a loss function, minimizing the error between the model predictions and the actual training data labels [17].

3) ***Model*** evaluation: Finally, the trained model is evaluated on a separate test dataset, which is not used during training or validation. This step provides a final measure of the model's performance on entirely new and unknown data, reflecting its actual ability to generalize and be applicable in real-world scenarios. The model evaluation may include performance metrics. In our case, we use the following metrics:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{1}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

where TP represents true positive, TN represents true negative, FP represents false positive, and FN represents false negative. Given this workflow, we aim to simplify and make it more accessible and educational for users by abstracting lines of code, leaving only the practical part of AI training.

B. Platform Operation

The platform was designed to simplify and facilitate artificial intelligence (AI) training to detect desired objects. Through six carefully crafted steps, users are guided through an intuitive process that spans from image capture to evaluation of the trained model.

The process begins by capturing images of the objects to be detected, allowing users to gather a representative dataset for model training. Next, users can split this dataset into the classic proportions of training, validation, and testing (70%, 15%, and 15%, respectively), allowing for a robust evaluation of the model's performance.

With the dataset prepared, the platform guides the user to perform precise annotations on the images through an external platform, ensuring the quality and accuracy of the training data. These annotations are stored along with the training images in a designated folder.

Before training begins, adjustments are made to the model parameters, such as the number of epochs and image dimensions, to ensure optimal configuration. With everything set up, training commences, and the user can monitor the model's progress in real time.

One distinctive feature of the platform is its ability to visualize the model in action through a webcam, providing practical and immediate experience. The results are displayed in real time, allowing users to assess the model's effectiveness while running.

Additionally, the platform automates the generation of performance metrics, such as precision, recall, and accuracy, on the basis of the results observed by the user. This enables an objective model evaluation and provides valuable insights for future refinements.

Fig. 2 visually illustrates the six steps of the AI training process provided by the platform, emphasizing the simplicity and effectiveness of the proposed workflow. This approach was designed to make AI training within the platform more accessible and understandable, even for users with no prior experience in machine learning, thus contributing to the democratization and widespread adoption of artificial intelligence across various domains.

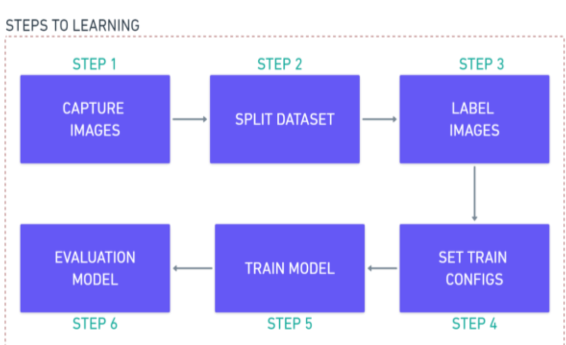

Fig. 2. Steps to learn.

IV. PLATFORM DEVELOPMENT

At the core, we have three main modules within the platform. Fig. 3 and Fig. 5 show these modules.

Within this final project conception, we have three modules containing the steps mentioned earlier to complete the learning process.

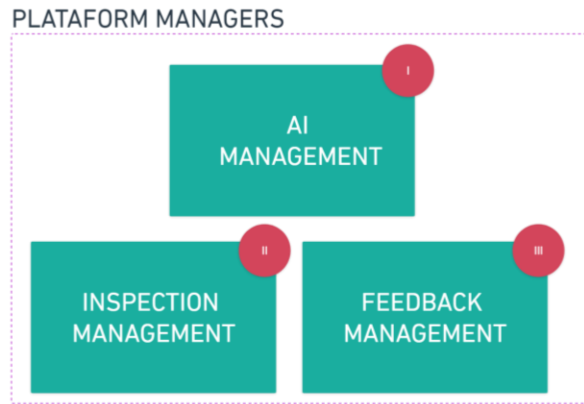

Fig. 3. Platform Management

A relational database, specifically MySQL, was utilized for better record-keeping and result control. The MySQL Workbench IDE was used to construct the database and tables. With this, a database was modeled to meet the following requirements:

- Which models were trained or uploaded to the platform to monitor and have the option of choosing the desired model?
- Results of the human evaluation
- What were the results of the inference, and what was desired to be obtained:
  - Image name.
  - Timestamp.
  - Result.
  - Model used.

In this context, Streamlit has a disadvantage: variables are reset when reloading or clicking a button. This means that we could not track which index of the image we were on or the value of our variable because clicking any button constantly reset it. To address this, control variables were created in the database. These variables were managed through updates in the database itself, and to read them, a select query was executed to retrieve their values. In this way, the variables retain the same value even with page refreshes. On this basis, the database in Fig. 4 was constructed.

The modules presented by the managers represent the different functionalities and stages of the YOLO network training process on the platform. Each module is designed to guide users through a specific part of the process, from dataset generation to model training and evaluation. The modules represent the steps.

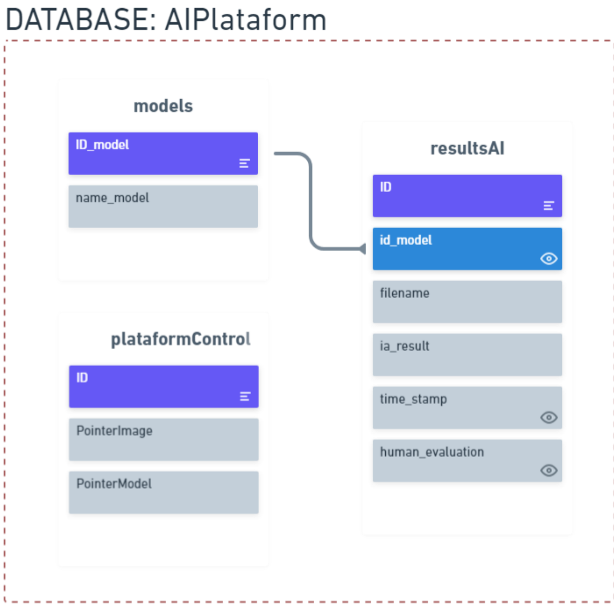

Fig. 4. Tables and database

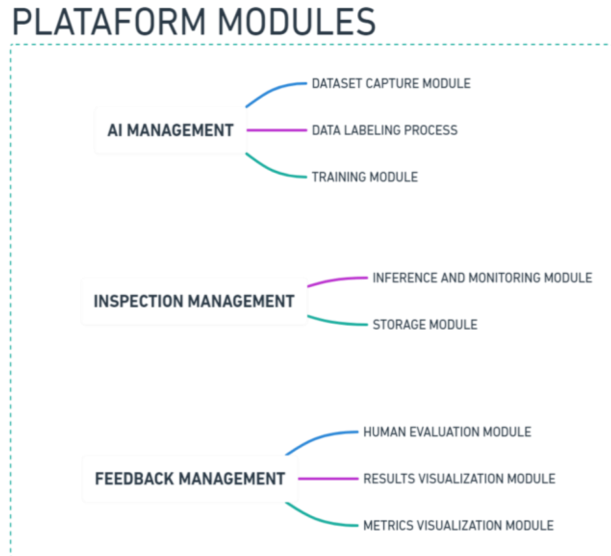

Fig. 5. Management module.

A. AI Management

1) **The AI training manager** is the central part of the platform, providing an intuitive and guided interface for users to follow the necessary steps to train a YOLO network and evaluate its performance. This module is designed to simplify the AI training process by dividing it into easily understandable and executable steps.

2) **Dataset Generation**: In this process stage, the platform allows users to capture images directly via an integrated camera. When the camera is activated, the platform automatically initiates image capture at 5-second intervals, which are saved to a designated folder that the user can create beforehand within the platform. This functionality significantly simplifies data generation for AI training, making it accessible and convenient. Users can build a representative and varied dataset by capturing images of the object of interest at regular intervals, which is essential for successful model training. Additionally, the platform provides valuable tips alongside the page to maximize training effectiveness. For example, it suggests capturing as many images as possible and varying lighting conditions, angles, and backgrounds to ensure comprehensive and robust learning by the model.

3) **Dataset Splitting**: After dataset generation, users can split the data into training, validation, and test sets, following the classic proportions widely used in machine learning practice. This step is critical to ensure that the model is trained, validated, and evaluated fairly and robustly, allowing for an accurate assessment of its performance. Splitting the dataset into separate sets enables an independent evaluation of the model on data unseen during training, providing valuable insights into its generalizability and applicability in real-world scenarios.

4) **Model training**: After dataset preparation, users can start training the model. Various model parameters can be adjusted in this phase to optimize the model performance. Among the most critical parameters are the number of epochs, which determines how many times the model iterates over the dataset during training, and the dimensions of the input images, which can significantly influence the model's ability to capture relevant details and patterns. This stage requires careful attention from users, as parameter choices can directly impact the quality and effectiveness of the final model.

B. Inspection Management

1) **Inference and Monitoring Module**: We can select the pretrained model from those in the model folder. With the chosen model, we can then activate our camera, and upon enabling it, we can see our model in action. The model automatically opens within the program and detects the object of interest.

2) **Storage** module: Each inference made at a 5-second interval is saved. Its image is stored along with its result in a database, where each image is associated with the model used to infer it and its result as the main characteristic in the database.

C. Feedback management

1) **Human evaluation and** results visualization module: This module displays the images and results so that a human evaluator can assess whether the inference has an adequate result. All this information is extracted from the database, where we have the name, the image saved, the AI result, and the model used in its inference, with the images all saved in a local folder on the disk. From this, the image, along with its result, will be displayed to an evaluator (human) so that they can classify it with one of the four possible AI responses:

- True positive.
- False positive.
- True Negative.
- False negative.

When one of these options is selected and the classification is made in the module, a previously empty column, which is the human evaluation column, is filled. Therefore, we obtained the AI results and human evaluations on the basis of these results, allowing us to generate metrics for model evaluation.

3) **Metrics Visualization Module**: With some human inspections performed, this module displays our model's precision and recall on the basis of the results of the previous module. The now-filled human evaluation column allows us to calculate these metrics. Here, we will also assess the number of inferences made per hour (and their results) and the percentage of each inference made throughout the day, displayed in a bar chart and a pie chart, respectively.

V. RESULTS AND CONCLUSIONS

The platform's objective is to assist in creating a new YOLO network model for detection. The platform thus functions as a guide and a way to abstract lines of code and various steps of library installation environment configurations, among other compatibility factors, so that all training can be successfully executed.

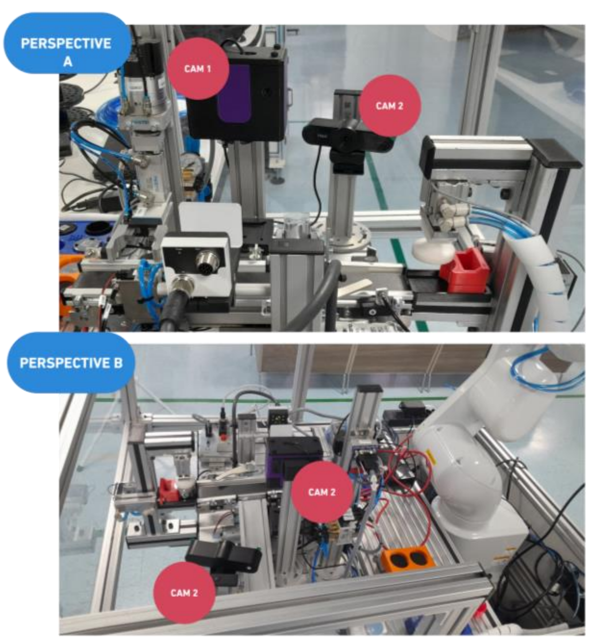

Fig. 6. Didactic plant: Measuring from Festo

A. Test Scenario

We created two models in a didactic plant that sealed a piece to assess the platform. In Fig. 6, it is possible to visualize two perspectives of the plant with two cameras. The didactic plant from the FESTO company simulates the sealing process of a lid with three distinct colors: pink, silver, and black. The plant executes two processes: initially, a gripper brings one of the pieces in front of a camera and then verifies whether the product is sealed. If it is sealed, the process directs the product to the output; otherwise, it is sealed by the robotic arm, which picks one of the lids and performs this process. After the sealing process, whether successful sealing was achieved was verified. Otherwise, it is directed to the press. After this, it is taken to the exit ramp, and if the robot seals correctly, it is also taken to the exit ramp.

Given this scenario, two networks were trained through the platform: one for identifying the sealed piece and the other for verifying whether the lid was correctly sealed.

D. Steps One and Two: Capturing and Splitting Datasets

In this step, we capture the images and split the dataset. Here, we can (1) create a new folder if the folder where you want to save the training images does not exist. Referring to Fig. 9.

Fig. 7. Images captured for training.

In section 2, the page can be created and then updated. Fig. 3 shows the camera selection process. A pressing start every 5 seconds will save the images within the previously selected folder. Section 4 divides our database into training, validation, and testing datasets. By default, it remains at 70–15–15. When dividing between training, testing, and verification, two additional folders are created within each folder called 'label' and 'images', where one contains the labels and the other contains their respective images so that it is possible to see images of the database in Fig. 7, following the YOLO notation standard. This organization can be seen in Fig. 8.

E. Step Three - Image Labeling

In the third step of the tutorial, users were guided through the process of using Makesense, an online platform for image annotation. They were instructed to upload their images to the platform and annotate objects of interest by marking bounding rectangles around them (see Fig. 10). After completing the annotation process for all images, users are directed to export the annotations as text files, usually in formats such as YOLO or similar formats, containing information about each annotated object's location and class.

These text files serve as labels for the corresponding images, providing essential training data for object detection models such as YOLO. Following these steps, users efficiently prepared their annotated pictures and labels for further processing in AI training pipelines.

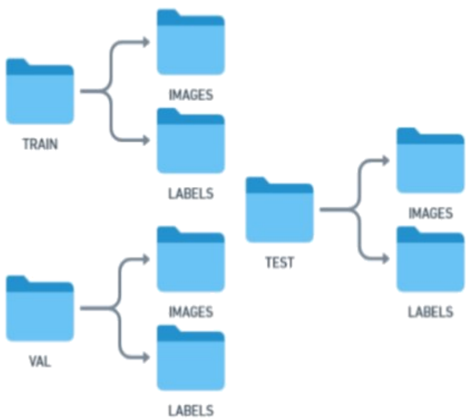

Fig. 8. Organization of Folders After Dataset Division.

F. Step Four and Five - Model Parameters and Training:

In this step, there are two processes. First, the directory where the respective labels of the images are located must be chosen, as shown in Figure 11, following the order. The model's name, number of epochs, and image dimensions can also be selected (Fig. 12). With all these parameters, model training can be initiated. Upon clicking the button for model training, a Python subprocess library is used to execute another Python script in parallel, specifically the training script. Once the training is complete, a message is displayed indicating successful completion, after which the new model is uploaded to the platform (see Fig. 12).

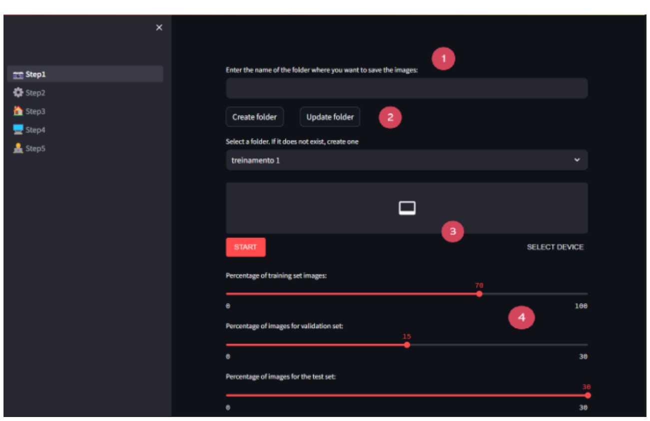

Fig. 9. Step one page.

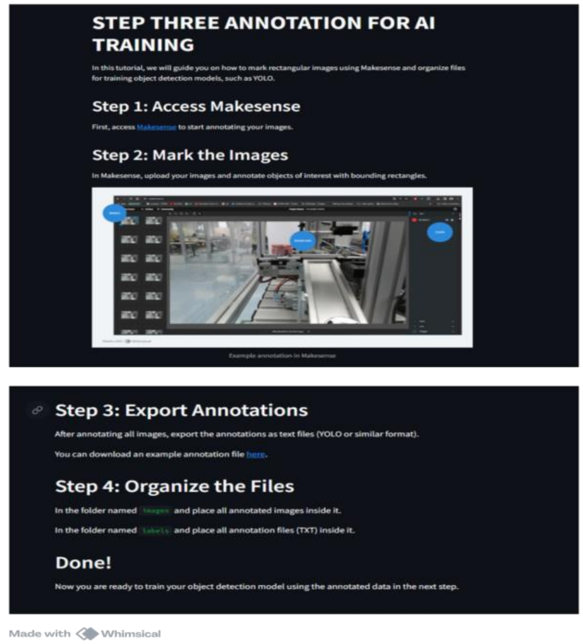

Fig. 10. Sep Three - How to use make sense.

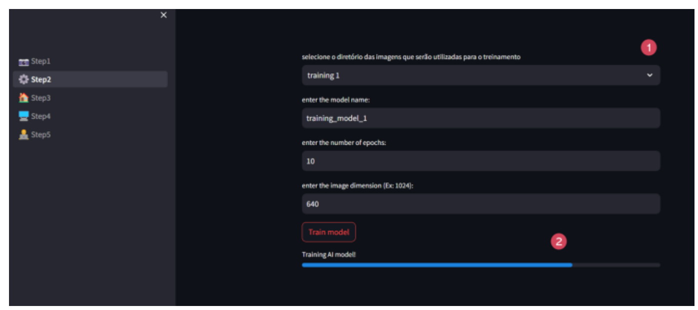

Fig. 11. Configuring training parameters

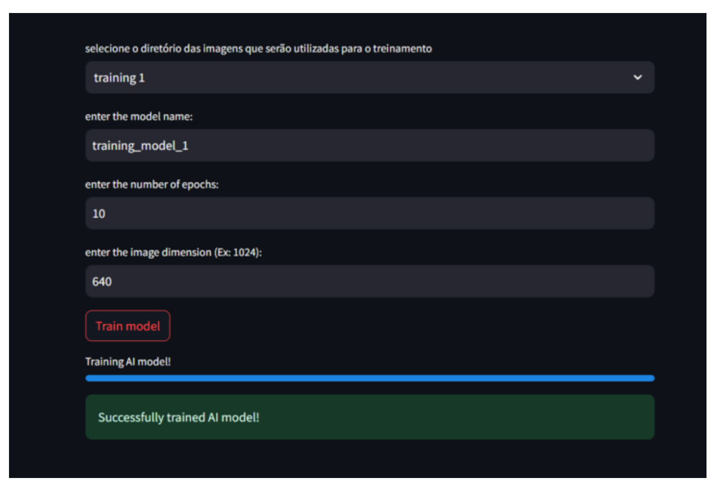

Fig. 12. Model training was completed.

There is then the option to upload the newly trained model; see Fig. 13. Once on the platform, this model is automatically registered in the database with the model's name in the file. The model can be monitored once in the database and once on the platform. In this experiment, two created models were evaluated: one for checking the sealing ability and the other for verifying whether the model was sealed correctly.

Fig. 13. The model was successfully uploaded to the platform.

E. Step six - Model evaluation

Models can be used in real time with the model trained and registered on the platform. On the main page, there is an option at the top to select the number of cameras to be used. If only one model is to be used, which is more common, the first option is set to 1. However, in this specific project, two models were used simultaneously, leading to the addition of this option. Figure 14 illustrates how the selection is made when only one camera is used.

In option 2, a single model can be selected for one camera. However, option 2 allows two cameras to choose a different model for each camera. The cameras can be modified in option 3 under 'Select Device.' In option 4, a common point between both scenarios is found, allowing the model selection to determine the quantity and specifics of the inferences per hour and day.

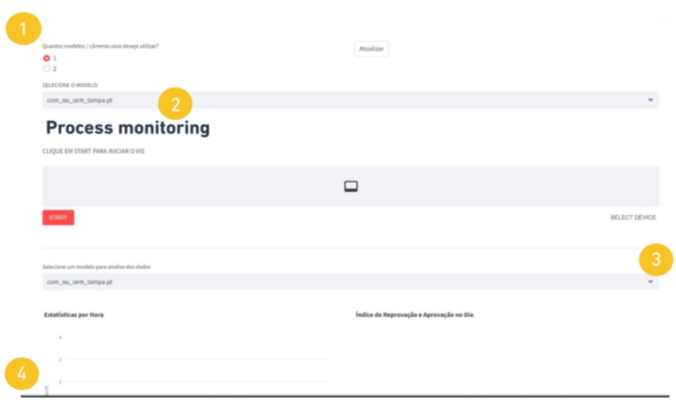

Fig. 14 Monitoring screen.

For example, regarding lids, we can observe the number of items passing through a plant with or without a lid per hour, along with the percentage of each throughout the day. This graph can be seen in Fig. 15, where the update button is located. This button refreshes the chart results, which are not updated in real time according to the inferences; a click is required to update them. Additionally, at position 2, we can see that we can change the model when we see the results.

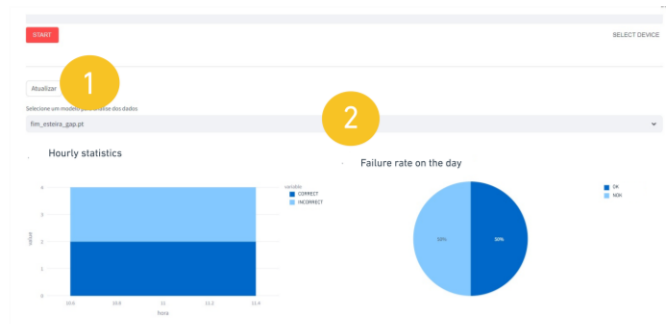

Fig. 15. Monitoring graphics.

ID	filename	ia_result	time_stamp	human_evaluation	model
316	1532024_142549....	INCORRETO	2024-03-15 14:25:49	NULL	com_ou_sem_tampa.pt
317	1532024_142550....	INCORRETO	2024-03-15 14:25:50	NULL	com_ou_sem_tampa.pt
318	1532024_142552....	INCORRETO	2024-03-15 14:25:52	NULL	com_ou_sem_tampa.pt
319	1532024_142553....	INCORRETO	2024-03-15 14:25:53	NULL	com_ou_sem_tampa.pt
320	1532024_142555....	INCORRETO	2024-03-15 14:25:55	NULL	com_ou_sem_tampa.pt
321	1532024_142556....	INCORRETO	2024-03-15 14:25:56	NULL	com_ou_sem_tampa.pt
322	1532024_142558....	INCORRETO	2024-03-15 14:25:58	NULL	com_ou_sem_tampa.pt
323	1532024_142559....	INCORRETO	2024-03-15 14:25:59	NULL	com_ou_sem_tampa.pt
324	1532024_142601....	INCORRETO	2024-03-15 14:26:01	NULL	com_ou_sem_tampa.pt
325	1532024_142603....	INCORRETO	2024-03-15 14:26:03	NULL	com_ou_sem_tampa.pt
326	1532024_142604....	INCORRETO	2024-03-15 14:26:04	NULL	fim_esteira_gap.pt
327	1532024_142605....	INCORRETO	2024-03-15 14:26:05	NULL	com_ou_sem_tampa.pt
328	1532024_142605....	INCORRETO	2024-03-15 14:26:05	NULL	fim_esteira_gap.pt
329	1532024_142606....	INCORRETO	2024-03-15 14:26:06	NULL	fim_esteira_gap.pt
330	1532024_142606....	INCORRETO	2024-03-15 14:26:06	NULL	fim_esteira_gap.pt
331	1532024_142607....	INCORRETO	2024-03-15 14:26:07	NULL	fim_esteira_gap.pt

Fig. 16. Data in the database.

Fig. 16 shows the responses saved in the database. During each of the inferences, images, and results, a human, namely, the platform user, evaluates whether the model's inference is correct. Assuming it is accurate or not. The human inspection page, shown in Fig. 17, displays the images inferred by the monitoring model, pulling from the table the name of the image (which is used to open the file). This table can be seen in position 5, the result of the human inspection, which, if done, appears to indicate the outcome; if not, it simply suggests that it has not yet been evaluated, in addition to the actual result from the AI, which is precisely what we will assess the model against.

At position 1, it is possible to filter by inferences that have not yet been evaluated, returning only images from the database where the human evaluation column is not empty. At position 2, the desired model can be selected. If all the models are enabled, images from both the "lid" model and the "gap" model will be available for evaluation, allowing for simultaneous assessment. However, if only a specific model is considered, it can be selected, and the images displayed for evaluation will be exclusively from the chosen model. At position 3, the inferred image can be viewed, with the image name, human inspection results, and AI outcome displayed above it for context. Finally, at position 4, the evaluation can be conducted according to the AI result.

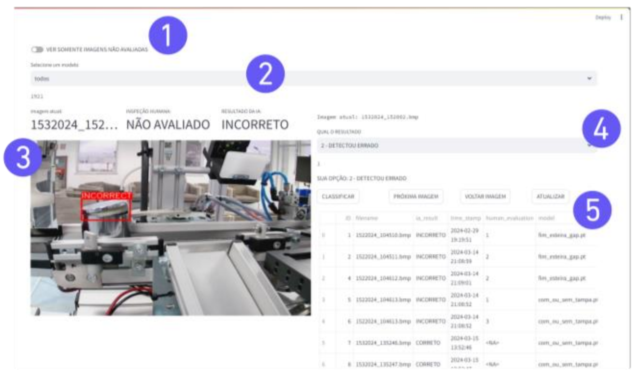

Fig. 17. Human evaluation screen

The evaluation system was based on the concept of a confusion matrix for AI assessment; hence, we have four options for classification:

1. Correct detection.
2. Incorrect detection.
3. Although it is detected, it does not exist.
4. They are not detected and do not exist.

When an image is evaluated, its evaluation column in the database is edited, changing from empty to the number indicating the evaluation.

After inspecting the images, it is essential to remember that not all photos need to be checked—only enough for the metrics to begin being generated. The dashboard screen can then be opened to evaluate the model's performance.

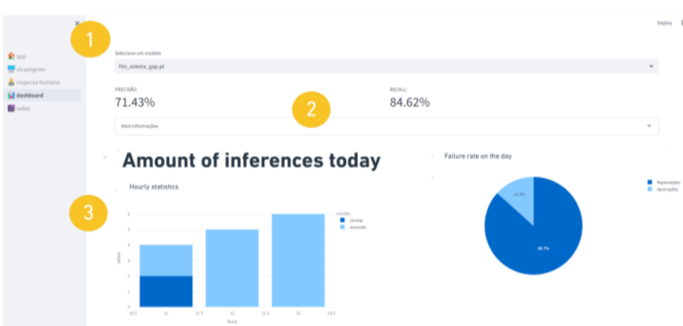

Fig. 18. Dashboard and metrics

In Fig. 18, the option to select the model for which the metrics are to be viewed is shown, along with the number of inferences per day at position 1. At position 2, the metrics calculated from the results generated by human inspection are displayed, with the main metrics being precision and recall. Accuracy was not used because only images detected by the model were analyzed; there were no images that the model failed to detect, which resulted in an accuracy of 100%.

F. Conclusion

This paper presents a platform that indicates, instructs, and facilitates the training of a convolutional neural network, specifically a YOLO model. Through just six steps, it is possible to capture images, label them, train the model, and execute it straightforwardly via a computer's webcam. This allows the newly trained model to be used in real time, with the ability to monitor its metrics. In an industrial context, detecting parts, faults, and anomalies is essential for optimizing processes and ensuring quality. The platform introduces users in a way that enables them to understand how AI training works and utilize this tool to foster innovation within a company or similar setting.

VI. NEXT STEPS

As platform development progresses, there is an opportunity to explore new areas for enhancement and expansion. One potential enhancement involves integrating advanced model evaluation techniques. This could include implementing more sophisticated methods, such as precision–recall and receiver operating characteristic (ROC) curves, which offer deeper insights into the model's performance across various scenarios.

Furthermore, the platform's training capacity should be expanded to allow users to train more complex and sophisticated models. This could involve supporting advanced neural network architectures and integrating their suggestions and feedback into the platform's future development. This will enable us to address users' specific needs and ensure that the platform meets their evolving expectations and requirements.

ACKNOWLEDGMENT

This research, carried out within the scope of the Samsung-UFAM Project for Education and Research (SUPER), according to Article 39 of Decree n° 10.521/2020 (SUFRAMA), was funded by Samsung Electronics of Amazonia Ltda., under the terms of Federal Law n° 8.387/1991, through agreement 001/2020, signed with the Federal University of Amazonas and FAEPI, Brazil. We also want to acknowledge the financial support from CAPES, CNPq, and FAPEAM.

References

[1] A. Ayub Khan, A. A. Laghari, and S. Ahmed Awan, "Machine Learning in Computer Vision: A Review," EAI Endorsed Scal Inf Syst, vol. 8, no. 32, p. e4, Apr. 2021.

[2] J. Zhang and D. Tao," Empowering Things With Intelligence: A Survey of the Progress, Challenges, and Opportunities in Artificial Intelligence of Things," in IEEE Internet of Things Journal, vol. 8, no. 10, pp. 77897817, May 15, 2021, doi: 10.1109/JIOT.2020.3039359.

[3] J. Bernal et al.," Deep convolutional neural networks for brain image analysis on magnetic resonance imaging: a review," *Artificial Intelligence in Medicine*, vol. 95, pp. 64-81, 2019.

[4] J. Lee, H. Davari, J. Singh, V. Pandhare," Industrial Artificial Intelligence for industry 4.0-based manufacturing systems," Manufacturing Letters, vol. 18, pp. 20-23, 2018, ISSN 2213-8463, DOI: 10.1016/j.mfglet.2018.09.002.

[5] M. Urgo, W. Terkaj, G. Simonetti," Monitoring manufacturing systems using AI: A method based on a digital factory twin to train CNNs on synthetic data," CIRP Journal of Manufacturing Science and Technology, vol. 50, pp. 249-268, 2024, ISSN 1755-5817, DOI: 10.1016/j.cirpj.2024.03.005.

[6] Ng, D.T.K., Lee, M., Tan, R.J.Y., et al. A review of AI teaching and learning from 2000 to 2020. Educ Inf Technol 28, 8445–8501 (2023). https://doi.org/10.1007/s10639-022-11491-w

[7] K. Bartsch, A. Pettke, A. Hubert, J. Lak¨ amper, and F. Lange, "On the¨ digital twin application and the role of artificial intelligence in additive manufacturing: a systematic review," Journal of Physics: Materials, vol. 4, no. 3, p. 032005, Apr. 2021, doi: https://doi.org/10.1088/25157639/abf3cf.

[8] J. Terven," A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS," MDPI, 2023.

[9] de Lucena Jr. V.F., De Queiroz-Neto J.P., Benchimol I.B., Mendonça A.P., Da Silva V.R., Ferreira Filho M. Teaching software engineering for embedded systems: An experience report from the Manaus research and development pole (2007) Proceedings - Frontiers in Education Conference, FIE, DOI: 10.1109/FIE.2007.4417966

[10] de Lucena Jr. V.F., Brito A., Gönner P., Jazdi N. A Germany-Brazil experience report on teaching software engineering for electrical engineering undergraduate students (2006) Software Engineering Education Conference, Proceedings, 2006, DOI: 10.1109/CSEET.2006.6

[11] Tyler Richards, Getting Started with Streamlit for Data Science: Create and deploy Streamlit web applications from scratch in Python, Packt Publishing, 2021.

[12] A Faster Way to Build and Share Data Apps. Accessed: Jun. 30, 2023. [Online]. Available: https://streamlit.io/

[13] Ultralytics, "Yolov8," 2020, access in: Jul. 2024. [Online]. Available: https://docs.ultralytics.com/pt/models/yolov8/

[14] T. Yarally, L. Cruz, D. Feitosa, J. Sallou e A. van Deursen, ”Uncovering Energy-Efficient Practices in Deep Learning Training: Preliminary Steps Toward Green AI”, 2023 IEEE/ACM 2ª Conferencia Internacional sobre Engenharia de IA - Engenharia de Soft-ˆ ware para IA (CAIN) , Melbourne, Australia, 2023, pp. 25-36, doi:´ 10.1109/CAIN58948.2023.00012.

[15] handrakala, CB, Bhardwaj, R. & Pujari, C. Um pipeline de reconhecimento de intenc¸ao para IA conversacional. Internacional j. informac¸˜ ao.˜ tecnologia. 16, 731–743 (2024). https://doi.org/10.1007/s41870-02301642-8.

[16] A. Bochkovskiy, C.-Y. Wang and H.-Y. M. Liao," YOLOv4: Optimal Speed and Accuracy of Object Detection", 2020, [online] Available: http://arxiv.org/abs/2004.10934.

[17] Kragh, M.F., Karstoft, H. Embryo selection with artificial intelligence: how to evaluate and compare methods?. J Assist Reprod Genet 38, 1675–1689 (2021). https://doi.org/10.1007/s10815-021-02254-6.