Synthetic Data-Driven Object Detection for Rail Transport: YOLO vs RT-DETR in Train Loading Operations

Thiago Leonardo Maria[1][a], Saul Delabrida[1,2][b] and Andrea Gomes Campos[1,2][c]
[1]*Graduate Program in Instrumentation, Control and Automation of Mining Processes (PROFICAM), Federal University of Ouro Preto (UFOP) and Vale Institute of Technology (ITV), Minas Gerais, Brazil*
[2]*Department of Computing (DECOM), Federal University of Ouro Preto (UFOP), Ouro Preto, Brazil*

Keywords: Wagon, Object Detection, Synthetic Data.

Abstract: Efficient wagon loading plays a crucial role in logistic efficiency and supplying essential raw materials to various industries. However, ensuring the cleanliness of the wagons before loading is a critical aspect of this process as it directly impacts the quality and integrity of the transported item. Early detection of objects inside empty wagons before loading is a key component in this logistic puzzle. This study proposes a computer vision approach for object detection in train wagons before loading and performs a comparison between two models: YOLO (You Only Look Once) and RT-DETR (Real-Time Detection Transformer), which are based on Convolutional Neural Networks (CNNs) and Transformers, respectively. Additionally, the research addresses the generation of synthetic data as a strategy for model training, using the *Unity* platform to create virtual environments that simulate real conditions of wagon loading. Therefore, the findings highlight the potential of combining computer vision and synthetic data to improve the safety, efficiency, and automation of train loading processes, offering valuable insights into the application of advanced vision models in industrial scenarios.

1 INTRODUCTION

Computer vision has emerged as a powerful tool in the industry, transforming the way object inspection and detection are carried out in industrial environments, all due to the exponential advancement of technology (Zhang et al., 2014). Combining image processing algorithms and artificial intelligence techniques, computer vision enables systems to interpret and understand the visual environment, enabling a variety of applications in industrial automation (Sonka et al., 2014).

Autonomy in industrial inspection is fundamental to guarantee the efficiency, quality, and safety of production processes. Traditionally, inspection is carried out manually, which is susceptible to environmental influences, often leading to human errors (El-Masry et al., 2012), in addition to requiring significant time and resources, and its accuracy is not guaranteed (Park et al., 1996). However, advances in computer vision have enabled automated inspection systems capable of analyzing and interpreting images quickly and accurately. These systems can identify defects, anomalies, and objects of interest in real time, all without requiring physical contact with the environment. (Aydin et al., 2014).

Object detection in industry is a key application of computer vision, with diverse applications including product tracking, automated selection, quality control, and workplace safety (Megahed and Camelio, 2012). The ability to detect and recognize objects in complex and dynamic industrial environments is essential to ensure process efficiency and prevent accidents and contamination. As a result, foreign object detection makes industrial processes more efficient and safer.

Rail transport plays an essential role in the global supply chain, providing a viable and sustainable alternative to road transport. In the United States and the European Union, the average volume of rail freight transported between 2006 and 2019 was approximately 2.8 trillion tonne-kilometers (Gholamizadeh et al., 2024). Statistics show that expanding rail networks and using them efficiently are crucial to meeting global logistics demand and supporting sustainable economic and environmental practices.

[a] https://orcid.org/0009-0002-8815-4989
[b] https://orcid.org/0000-0002-8961-5313
[c] https://orcid.org/0000-0001-7949-1188

Maria, T. L., Delabrida, S. and Campos, A. G.
Synthetic Data-Driven Object Detection for Rail Transport: YOLO vs RT-DETR in Train Loading Operations.
DOI: 10.5220/0013402500003929

In *Proceedings of the 27th International Conference on Enterprise Information Systems (ICEIS 2025) - Volume 1*, pages 873-880
ISBN: 978-989-758-749-8; ISSN: 2184-4992

Railroads often pass through areas where low-income people live. Due to the presence of single-track sections of a single line where there is traffic in both directions, the train must wait to pass through the route it chooses. Therefore, trains that are not operating must be stopped at intersections and in areas adjacent to cities. It is not uncommon for waste to be improperly discarded into the wagons, posing challenges to operational efficiency and cargo integrity

Early detection of objects inside these empty wagons before loading is a critical necessity to ensure operational efficiency and safety of the logistics process. The presence of unwanted objects can compromise not only the logistical organization but also the integrity of the transported item.

In the mining industry, trains must typically maintain a speed of approximately 10 km/h when entering a loading terminal to perform the tare process. However, when foreign object detection process is carried out visually by the operator, this speed is reduced in some terminals. Consequently, the tare time is increased, which leads to a delay in the start of charging.

The relevance of keeping wagons clean before loading goes beyond logistical organization. Cleaning plays a crucial role in preserving the quality of the transported item, especially in the mining industry, preventing unwanted contamination that could compromise the integrity of the final product. Efficient object detection inside wagons not only optimizes the loading process but also ensures the quality of the ore from its origin to its final destination, affecting both its value and reliability.

It is often difficult to obtain real data for training object detection models. Some of these difficulties may be dangerous or difficult-to-access environments, rare or extremely specific scenarios, applications with private or sensitive data, cost of data collection, and ethics and consent for data collection. Many of these data acquisition difficulties come with the search for synthetic data, and one way to obtain this data is to use AI to generate this data. However, this method comes with some difficulties, some of which are the lack of perfect realism, it requires a lot of computational effort, it can generate bias in the generated data, problems with dynamic scenarios, and difficulty in generating extreme cases. Another alternative for generating synthetic data is to create a realistic virtual environment that simulates the situation from which you want to obtain the data.

The main objective of this study is to develop a computer vision-based approach for detecting objects in train wagons before loading operations.

The specific objectives include:

- Address the generation of synthetic data as a strategy for training the models, using the *Unity* platform to create virtual environments that faithfully simulate the real conditions of wagons.
- Compare two object detection models, one widely recognized, YOLO (You Only Look Once) v.10 and RT-DETR (Real Time Detection Transformer), based on convolutional neural networks (CNNs) and Transformers, respectively, highlighting their particularities and advantages for the detection of objects specifically in train cars.
- Contribute to the efficiency and safety of the wagon loading process, exploring the capabilities of computer vision, the comparison between established models, and the generation of realistic synthetic data.

Using computer vision and comparing two known models trained using realistic synthetic data generated by 3D simulation, this paper aims to analyzing which is the best specific object detection system for train loading with that increasing the efficiency and safety in train loading by assisting the loading operator in decision-making. Progress made in this area not only improves operational practices but also encourages a smarter and more technologically advanced approach to rail transportation and logistics.

2 RELATED WORKS

The use of computer vision and artificial intelligence can solve many practical problems, including those currently being solved in the industrial environment. This chapter examines several ways in which these technologies can be used to detect foreign objects and anomalies in the industrial environment.

(Yang et al., 2014) presents a real-time inspection system for conveyor belts using computer vision. The objective of the system is to find and classify unwanted objects that may impair the efficiency or safety of the conveyor process in industrial production lines. The proposed system analyzes the images captured by cameras installed along the conveyor belt and, using a combination of machine learning and image processing methods classifies the objects to make decisions. The experimental results show the effectiveness of the proposed system in detecting and classifying objects in real-time, highlighting its potential to automate and improve inspection processes in industrial production lines.

(Yao et al., 2023) present an improved method for detecting foreign objects on conveyor belts using the YOLOX model. To detect unwanted objects on industrial conveyor belts, they introduced modifications to

the YOLOX model. This improves industrial inspection systems by ensuring the safety and efficiency of transportation processes. But this system often needs adjustments for different belt types, materials or defect patterns. This limits generalization and requires specific settings for each application environment.

In the railway (Chen W, 2022) describes a system for detecting foreign objects in images, using a two-stage convolutional neural network. The authors propose an approach to identify and classify unwanted objects in railway track images, aiming to improve the safety and efficiency of railway operations. The system is trained on a large dataset of railway images, allowing it to learn to detect and classify foreign objects with high accuracy.

(Hütten et al., 2022) explores the application of the Transformer architecture in industrial visual inspection, trained in three different use cases in the domain of visual damage assessment for railway freight car maintenance. This paper explains how the Transformer approach has been modified and used for visual inspection tasks. The paper discusses the advantages and disadvantages of the ViT approach compared to other traditional convolutional neural network architectures, such as CNNs. The experiment results show that the Vision Transformer architecture can be competitive in several industrial visual inspection tasks, outperforming or equaling other conventional architectures.

(Salas et al., 2020) presents an approach to train object detection and segmentation models on images of industrial machinery. Rather than relying solely on real-world datasets, the authors propose using computer-generated synthetic images to train and improve the performance of models. The paper describes the process of synthetic image generation, which involves creating 3D models of industrial machines and objects of interest, rendering these models in different virtual environments, and capturing images from these environments. After generating the synthetic images, the authors conduct experiments to compare the performance of models trained with synthetic data compared to models trained only with real data.

Most of the articles presented use CNNs in their object detection, (Yang et al., 2014) and (Yao et al., 2023) mention some limitations of using this type of approach on conveyor belts, such as adverse environmental conditions, such as dust, variable lighting and complex background noise. These factors can affect the detection accuracy and robustness of the model, especially in scenarios with poor visibility or partially obscured objects. (Hütten et al., 2022) validates the use of the Transformer architecture saying that it has a better performance than other traditional architectures in industrial visual inspection, this highlights the potential of the Transformer approach to boost automation and efficiency in industrial environments through advanced visual inspection. (Salas et al., 2020) demonstrate that training with synthetic data, in addition to real data, can lead to significant improvements in the accuracy and robustness of the models, especially in challenging conditions that were not adequately covered by real data.

3 METHODOLOGY

The methodology of this work consists of the following steps: 3D modeling and texturing, scene creation in Unity, synthetic data generation, training YOLO and RE-DETR detection networks, and comparing results.

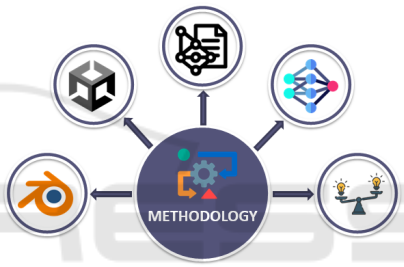

Figure 1: Steps of the proposed methodology.

3.1 3D Modeling and Texturing

3D models are necessary for the creation of the simulated 3D virtual environment. The models used in this work will be acquired from marketplaces. The marketplaces used were blendswap.com, free3d.com, assetstore.unity.com and mixamo.com. Most of the objects were obtained for free and some that could not be found this way were purchased. Models not found this way will be modeled and textured using the *Blender* tool, which is a free modeling tool.

The objects will be divided into two categories: main objects and foreign objects. The main objects will be all the objects that make up the scene, such as the train car, the tracks, vegetation, and terrain. The foreign objects will be the unwanted objects that will appear inside the cars. The foreign objects will be divided into two categories: inanimate objects and living objects: animals and humans. A total of 200 objects and variations will be selected, with 100 inanimate objects, 50 humans, and 50 animals.

Inanimate objects and animals will be chosen according to the chance of this type of object appearing

inside the wagons and the quantity of objects chosen to provide sufficient variability in the generation of synthetic data. The human category will have 10 different humans in 5 different poses.

Figure 2: Example of objects, animal and human category respectively.

3.2 Scene Creation in Unity

Unity is a game engine widely used for simulations and creation of interactive virtual environments.

The construction of the simulated scene in *Unity* is a crucial step, it consists of assembling a virtual environment capable of simulating the reality of train loading with a certain realism, using acquired and modeled objects incorporating various techniques to increase the fidelity of the simulation, techniques such as:

- Using ray tracing to enhance visual realism, allowing for a more accurate representation of light-object interaction in the scene (Skala et al., 2024).
- Development of a dynamic lighting system, capable of adjusting to the variable conditions of the simulated environment, providing realistic variations in luminosity.
- Implementation of a rain system to add climate variations to the simulation.
- Addition of wagons representing mineral waste to incorporate realistic conditions.
- Implementation of a random arrangement of objects inside the wagons to increase the variability of the synthetic data.

The *Unity* tool has three rendering pipelines, one of which is the High Definition Render Pipeline (HDRP), which prioritizes graphic quality using several tools. One of these tools is ray tracing, a technology that simulates the behavior of light in the real world to create more realistic and immersive graphics. The use of this pipeline also facilitates the creation of a climate system with rain cycles and has many post-processing tools that allow us to add effects to the camera to improve the visuals.

Unity uses C# to create scripts, these scripts allow the manipulation of almost everything within the tool. With these scripts, the entire dynamic system of this simulated scene will be developed, such as the movement of the trains, the day and night cycle, climate control, and the random appearance of objects inside the wagons, among other tasks.

3.3 Synthetic Data Generation

From the virtual loading environment previously created in *Unity*, with all simulations and techniques to increase realism, some additional configurations will be made to generate synthetic data. This process will be conducted following the steps:

- Careful selection of camera settings, including resolution, aspect ratio, FOV, and position/tilt, aligned with the usual specifications in the industry.
- Configure post-processing effects such as bloom, ambient occlusion, motion blur and vignette.
- Creation of a script to generate tag files to label objects in the simulated scene, facilitating the training of detection models.
- Generation of images to compose the training and validation data set, considering a comprehensive representation of the object detection conditions.

Will be generated a set contain 6600 images, 5280 for training and 1320 for validation, with 10% of background images in each category. These images will contain different settings such as day, night, day with rain and night with rain. This number of images was calculated based on *Ultralytics* suggestion in its documentation.

3.4 Training YOLO and RE-DETR Model

The training phase of the YOLO and RT-DETR detection networks involves preparing the synthetic data generated. This step will be performed using training and testing sets from the *Unity* simulation of the different experiments, with the appropriate object annotations. For this training, the RT-DETR-l and RT-DETR-x models will be used, which are the large and extra large models, respectively. In addition, the YOLO models to be used are YOLOv10l and

YOLOv10x, which are compatible with the RT-DETR models used. These models were chosen according to the availability of models provided by *Ultralytics*.

The training will be performed on a physical machine equipped with a 12-core AMD Ryzen 9 5900X processor, an NVIDIA GeForce RTX3090 graphics card with 24GB of VRAM and 64GB of DDR4 RAM at 3200MHz.

3.5 Comparing Results

The final phase of the research involves a comparison between the YOLO and RT-DETR networks considering the different experiments.

The MAP (Mean Average Precision) metric will be used, which is commonly used to evaluate the accuracy and performance of object detection algorithms in computer vision. It is especially relevant in object detection tasks where multiple objects can be present in a single image and it is important to evaluate the detection accuracy for each object class.

The mAP50 and mAP90 metrics will be used to evaluate object detection models. Evaluation at Different Accuracy Thresholds mAP50 measures model performance considering a more relaxed intersection-over-union criterion. It is useful for evaluating whether the model can detect objects at approximate positions, even if the segmentation or bounding is not highly accurate. mAP90 measures performance under a more rigorous criterion. It focuses on detection accuracy, evaluating whether the bounding boxes are perfectly aligned with the objects.

4 RESULTS

During the course of the project, several versions of the synthetic data generator were developed. The first was the proof of concept, in which a model of moving wagons was created, in which objects such as cubes and spheres were randomly inserted, replicating the conditions found in a railway environment. These objects were later replaced by items found inside wagons, increasing the fidelity of the simulation.

To control the movements of the wagons and the items that appear inside them, a dedicated script was developed. This script allows the variation of parameters such as the speed of the wagons and the chance of a wagon having an item inside. In addition, the strategic addition of a camera positioned above the wagons allowed for the accurate capture of images, essential for object detection and subsequent analysis.

As an integral part of the simulation, a dynamic day-night cycle system was implemented, contributing to diversifying the lighting conditions and increasing the realism of the scene. This approach enriches the simulated environment, making it more complex and closer to the real conditions encountered during loading operations.

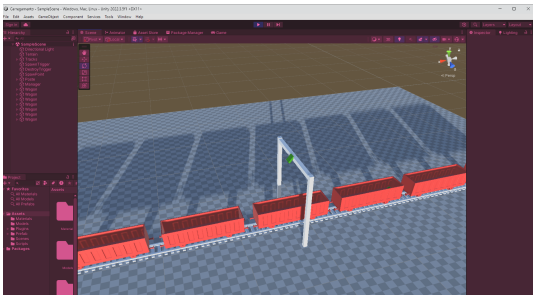

Figure 3: Proof of concept scene in *Unity*.

4.1 Acquisition of 3D Objects

The next step was to obtain the 3D objects. Scene composition objects such as train tracks, trees, and terrain were modeled and textured in the *Blender* tool. The train car, one of the main models requiring more details, was acquired from a 3D object marketplace and subsequently applied a custom texture using the *Blender* tool.

The models of inanimate objects, animals, and humans were mostly acquired for free from various marketplaces; all models have a CC-00 license.

All models went through an adaptation process to be used in *Unity*, which consisted of reducing the mesh size for objects with a very high polygon count and retexturing for some objects.

4.2 Configuration Script

After collecting the 3D models, the development of the synthetic data generator began. Based on the proof of concept, new 3D models were added, creating a more realistic virtual environment.

A script called “GameManager” was created to manage and configure all the settings for the symmetric data generator. This script has multiple settings that allow for the customization of the loading environment, the objects that appear randomly inside the wagon, and the dynamic weather and climate conditions. Additionally, the script automates the creation of the folder structure needed to separate and save the generated data.

4.3 Positioning of Objects

The positioning of the objects inserted into the wagon is randomly chosen from eight positions along the

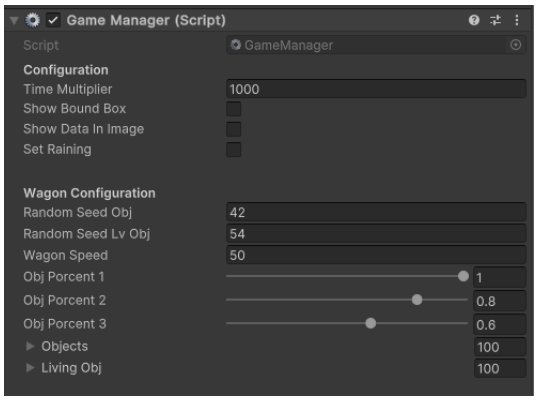

Figure 4: Part of the configuration script.

wagon, in addition to the rotation of the object on the vertical axis being defined randomly. Objects are placed slightly above the wagon, and the *Unity* physics engine simulates the item falling into the wagon. This process allows objects to rotate and settle in various orientations, exponentially increasing the diversity of their positions within the wagon.

The number of objects inside the wagon ranges from 1 to 3. The first object will always be inserted, the second object has a 50% chance of appearing, and the third object has a 25% chance, ensuring variability in the dataset while maintaining a realistic scenario.

4.4 Day and Night and Rain Cycles

The day and night cycle implemented in the proof of concept uses *Unity*'s lighting systems to configure light emission, activate dynamic shadows, and rotate sun objects.

In addition to the day and night cycle, a rain cycle was implemented. This rain cycle uses *Unity*'s particle system to simulate rain and has a random intensity of rain, fog, and lighting every time the rain starts. Furthermore, textures for the wagon, environment, and objects are dynamically modified to appear wet when the rain cycle starts.

To enhance the realism of the simulated environment, Ray Tracing and other filters such as Bloom, Screen Space Ambient Occlusion, Screen Space Reflection and several others were used. These filters are commonly used in games with realistic graphics, which helps maintain fidelity to real environments. The combination of these filters with day and night and rain cycles helps to diversify the synthetic data generated.

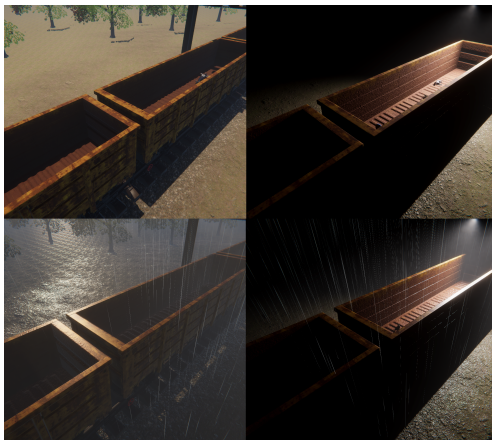

Figure 5: The cycles in the simulation.

4.5 3d Object Separation

In addition to the objects, humans and animals division, another general division of the 3D objects is made. They are randomly divided into five groups using a seed to enable repeatability. This division is necessary so that the training data and validation data contain different 3d models, promoting better generalization of the trained models. Four of these five groups will be used for training and the last for validation.

4.6 Label File and Synthetic Data Generation

One of the advantages of using synthetic data is the ability to generate labels automatically. In *Unity* we used the list of vertex positions of the mesh of the objects inside the wagon. This list of vertex positions is collected and converted to the camera space. By finding the maximum and minimum horizontally and vertically, the bounding box of the object can be determined. After that, it is necessary to apply some limiting filters so that the values do not exceed the screen limits and do not fall behind the wagon wall. With the label data collected, all that remains is to convert them to the label system that will be used. In this case, the YOLO system will be used.

To generate synthetic data, it is checked whether there are objects within the viewing space of the camera that is inside the wagon. If there are any objects, an image is generated and the process of generating the label for this image begins. After that, the image and the label are saved in their respective folders. After this, a period of time is waited for the next iteration and generation of the next data, this process is repeated until the defined amount of data is reached.

Figure 6: Example of data generated.

A total of 5.280 training images were generated, including 480 of which are background-only images and 1.320 validation images, 120 of which are background-only images. The images have a resolution of 1.080x1.080 pixels and were generated in approximately 6 hours. Figure 6 shows an example with the bounding boxes already placed on the generated data.

4.7 Model Training

Before the main model training, some pre-training phase was performed to define the training resolution of the images and the ideal number of epochs for this dataset. Four training sessions were performed with resolutions of 1.080p, 640p, 480p and 384p in 50 epochs. The results are shown in the Table 1.

Table 1: Resolution pre-training result.

	mAP50	mAP50-90	Train Time (h)
1080p	0.811	0.573	2.28
640p	0.809	0.561	0.85
480p	0.800	0.551	0.53
384p	0.765	0.523	0.38

After analyzing the results, the training resolution was defined as 480p, since higher resolution values did not show much improvement in the mAP50 values and at 480p the training time was reduced considerably compared to higher resolutions, the inference time will also be reduced in lower resolution. These tests were performed for both YOLO and RT-DETR, obtaining comparable results.

Another pre-training session was performed to define the number of epochs for training the models.

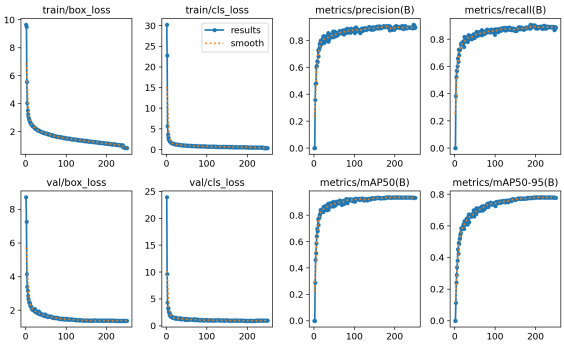

Figure 7: Epoch pre-training result.

Training was performed without an epoch limit with a patience of 50, that is, 50 epochs without any improvement and the training ended. The training was completed in 246 epochs. Analyzing the loss and mAP graphs in the Figure 7, a limit of 250 epochs was defined for training.

After the training configurations were defined, the synthetic data generated was separated, the fifth folder was selected as validation and the other four folders was selected as training. Subsequently, the four models, YOLOV10l, YOLOV10x, RT-DETR-l and RT-DETR-x were trained using the models provided by the *Ultralytics* library. Each training session took an average of 6 hours to complete. The results are presented in the Table 2 and Table 3.

Table 2: mAP50 values for classes and models.

Model	All	Obj	Animal	Human
YoloV10l	0.902	0.890	0.832	0.981
YoloV10x	0.909	0.890	0.855	0.982
RT-DETR-l	0.817	0.776	0.717	0.956
RT-DETR-x	0.800	0.747	0.702	0.950

Table 3: mAP50-95 values for classes and models.

Model	All	Obj	Animal	Human
YoloV10l	0.727	0.773	0.593	0.814
YoloV10x	0.729	0.765	0.612	0.811
RT-DETR-l	0.620	0.637	0.470	0.755
RT-DETR-x	0.589	0.595	0.435	0.736

The two tables provide comparisons between the YoloV10l, YoloV10x, RT-DETR-l, and RT-DETR-x detection models on different performance metrics, mAP50 (Table 2) and mAP50-95 (Table 3). On mAP50, YoloV10x performed best overall with 0.909, beating YoloV10l with 0.902. The RT-DETR models have lower values, with RT-DETR-x achieving 0.800 and RT-DETR-l 0.817.

In the mAP50-95 metric YoloV10x continues to lead with 0.729 in overall performance, closely fol-

lowed by YoloV10x with 0.727. Although YoloV10l and YoloV10x maintain high and similar performance, the RT-DETR models face considerable difficulties with the mAP50-95 metric, indicating that they have lower generalization ability across multiple accuracy thresholds.

5 CONCLUSIONS

This work developed a synthetic data generator within a fully virtual environment created using the *Unity* tool. The environment was created specifically before the loading of trains to evaluate two computer vision models, one very well-known and more traditional that uses convolutional neural networks, YOLO, and another model that is based on Vision Transformers (VITs). Specifically in this work, RT-DETR is being used, which is a real-time VIT.

From the training results of the four models, two from YOLO and two from RT-DETR of compatible sizes, it was observed that, for this environment, in the context of train loading, YOLO outperforms RT-DETR by approximately 10% mAP results. Additionally, no significant improvement was seen in performance with the larger model sizes of either YOLO or RT-DETR, indicating that increasing the model size did not yield better results for this specific task.

In a real-world scenario, the insights from this study show that a YOLO network can be applied in the area of train loading to improve automation and monitoring processes in loading operations. The ability of computer vision models to detect and analyze loading conditions in real time can significantly improve efficiency and become a great ally for the operator reducing human error and increasing safety.

Grounds for future work include expanding the analysis to other vision networks, comparing performance on different hardware, and comparing the use of synthetic data, real data, and mixed data in training the networks.

ACKNOWLEDGEMENTS

The authors thank the entire team of the Master's Program in Instrumentation, Control and Automation of Mining Processes (PROFICAM), Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG), Vale Technological Institute (ITV) and Federal University of Ouro Preto (UFOP). This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001, the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPQ) financing code 306101/2021-1, FAPEMIG financing code APQ-00890-23 and APQ-01306-22, the Instituto Tecnológico Vale (ITV) and the Universidade Federal de Ouro Preto (UFOP).

REFERENCES

Aydin, I., Karakose, M., and Akin, E. (2014). A new contactless fault diagnosis approach for pantograph-catenary system using pattern recognition and image processing methods. *Advances in Electrical and Computer Engineering*, 14(3):79–89.

Chen W, Meng S, J. Y. (2022). Foreign object detection in railway images based on an efficient two-stage convolutional neural network. *Comput Intell Neurosci*.

ElMasry, G., Cubero, S., Moltó, E., and Blasco, J. (2012). In-line sorting of irregular potatoes by using automated computer-based machine vision system. *Journal of Food Engineering*, 112(1):60–68.

Gholamizadeh, K., Zarei, E., and Yazdi, M. (2024). *Railway Transport and Its Role in the Supply Chains: Overview, Concerns, and Future Direction*, pages 769–796. Springer International Publishing, Cham.

Hütten, N., Meyes, R., and Meisen, T. (2022). Vision transformer in industrial visual inspection. *Applied Sciences*, 12(23).

Megahed, F. and Camelio, J. (2012). Real-time fault detection in manufacturing environments using face recognition techniques. *Journal of Intelligent Manufacturing*, 23:1–16.

Park, B., Chen, Y.-R., Nguyen, M., and Hwang, H. (1996). Characterizing multispectral images of tumorous, bruised, skin-torn, and wholesome poultry carcasses. *Transactions of the ASABE*, 39:1933–1941.

Salas, A. J. C., Meza-Lovon, G., Fernández, M. E. L., and Raposo, A. (2020). *Training with synthetic images for object detection and segmentation in real machinery images*. IEEE.

Skala, T., Maričević, M., and Čule, N. (2024). Optimization and application of ray tracing algorithms to enhance user experience through real-time rendering in virtual reality. *Acta Graphica*, 32(2):108–129.

Sonka, M., Hlavac, V., and Boyle, R. (2014). *Image Processing, Analysis, and Machine Vision*. Cengage Learning.

Yang, Y., Miao, C., Li, X., and Mei, X. (2014). On-line conveyor belts inspection based on machine vision. *Optik*, 125(19):5803–5807.

Yao, R., Qi, P., Hua, D., Zhang, X., Lu, H., and Liu, X. (2023). A foreign object detection method for belt conveyors based on an improved yolox model. *Technologies*, 11(5).

Zhang, B., Huang, W., Li, J., Zhao, C., Fan, S., Wu, J., and Liu, C. (2014). Principles, developments and applications of computer vision for external quality inspection of fruits and vegetables: A review. *Food Research International*, 62:326–343.