MATEC Web of Conferences **406**, 10003 (2024) https://doi.org/10.1051/matecconf/202440610003
2024 RAPDASA-RobMech-PRASA-AMI Conference

Robotic waste sorting using deep learning

Yurisha Goorun[1*], *Stephen* Marais[2], and *Fred* Nicolls[3]

[1]Centre for Robotics and Future Production, CSIR, South Africa
[2]Centre for Robotics and Future Production, CSIR, South Africa
[3]Department of Electrical Engineering, University of Cape Town, South Africa

Abstract. The ongoing waste crisis has highlighted the need to address the problem of waste collection and sorting. This paper presents an automated waste sorting system with Deep Learning using a stereoscopic camera and a robotic manipulator. The system can recognise and classify different types of waste and sort them into specific categories by using the robotic arm to improve the recycling process.

1 Introduction

Effective waste management is a critical challenge many countries face worldwide, and South Africa is no exception. With a population exceeding 60 million, South Africa generates significant waste annually, necessitating robust and efficient waste management systems [1]. Unfortunately, traditional waste segregation methods are predominantly manual, leading to inefficiencies, increased labour costs, and higher error rates [2]. These shortcomings highlight the need for automated solutions to enhance the accuracy and efficiency of waste sorting processes.

The improper handling and segregation of waste pose severe environmental and health risks [2]. Inadequate waste management systems can lead to pollution, the spread of diseases, and the inefficient use of valuable resources [3]. Moreover, manual waste sorting is labour-intensive and often subject to human error [4], resulting in lower recycling rates and higher contamination levels of recycled materials. The global push towards sustainable development highlights the necessity for advanced technologies to improve waste management practices and reduce the environmental footprint.

Artificial intelligence and robotics advancements have paved the way for innovative solutions to complex problems in recent years. Deep learning, a subset of machine learning, has shown remarkable success in various applications and object detection. Deep learning offers a promising approach to automating waste segregation combined with robotic arm manipulation. By leveraging state-of-the-art algorithms, robotic systems can be trained to accurately identify and classify different types of waste and perform precise pick-and-place actions, thus streamlining the waste management process.

* Corresponding author: ygoorun@csir.co.za

Automating waste sorting processes can significantly improve efficiency, reduce operational costs, and mitigate the environmental impact of waste. The deployment of advanced deep-learning algorithms enables the system to handle diverse waste types, ensuring high accuracy in identification and classification. Additionally, robotic arms can be programmed to adapt to various waste forms and sizes, enhancing the versatility and reliability of the sorting process.

This research aims to develop an efficient and accurate automated waste sorting system that integrates deep learning for object detection and robotic manipulation for waste handling. The YOLO (You Only Look Once) [5] algorithm, known for its speed and precision, performs real-time object detection. The stereoscopic ZED2i camera provides depth information, which is essential for accurately estimating the detected waste. The UArm Swift Pro robotic arm executes the pick-and-place operations based on the detected waste category.

2 Methodology

This methodology outlines a streamlined approach to developing an automated waste sorting system. It integrates advanced technologies to achieve object detection, position estimation, and efficient robotic manipulation for waste segregation. This solution has 3 three core stages: Object Detection, Object Position Estimation, and Robot Manipulation. The system uses ROS2 (Robotics Operating System) to communicate between components. Figure 1 shows a high-level system overview.

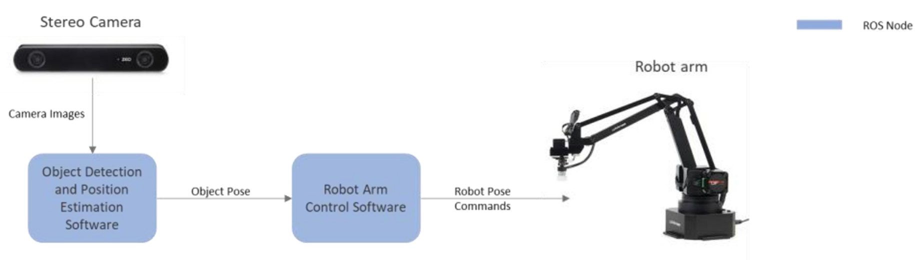

Fig. 1: High-level system overview.

2.1 Object detection

2.1.1 Data collection and preparation

State-of-the-art object detection algorithms require high-quality datasets. The Trash Annotations in Context (TACO) dataset [6] is the benchmark dataset for waste detection. It currently contains 1500 images with 4784 annotations and images belonging to 60 different categories. For this study, the TACO dataset was simplified into two categories, paper and plastic, to align with the waste collection system's focus.

2.1.2 Model training

This study uses the state-of-the-art model YOLO to perform object detection. YOLO uses a single Convolutional Neural Network (CNN) to predict multiple two-dimensional (2D) bounding boxes and class probabilities for those boxes simultaneously, in contrast to traditional methods that use a pipeline of stages. The algorithm also predicts the class and the probabilities for each bounding box. These probabilities indicate the likelihood of the object belonging to each predefined category: paper or plastic. Due to its architecture, YOLO is extremely fast, which makes it suitable for real-time applications such as waste detection. Both YOLOv5 [7] and YOLOv8 [8] models are used and compared in this study. Each model is trained for 100 epochs, with a batch size 32 and a learning rate of 0.01.

The dataset is, by default, in COCO (Common Objects in Context) format. To use it, the data format required by YOLO must be converted to a YOLO-accepted format. An online tool, Roboflow [12], converts different data formats for specific detection models, which is used for this task.

2.2 Object position estimation

2.2.1 Depth sensing

Leveraging the stereoscopic ZED2i camera, 3D information can be obtained from the detected objects. The range at which objects are detected is within the camera parameters of 0.5m to 20m. The depth data obtained is essential for estimating the object's position as it gives the z dimension of the object's position.

2.2.2 Synchronisation with object detection

The information from the 2D boxes, obtained from the object detection stage, is used to compute the object's centre point (x_{centre}, y_{centre}) in the camera frame. The centre point of the object is looked up in the depth image to find the corresponding z_{centre} point, which represents the depth value. This process provides the coordinates of the object in the camera frame. To convert the 2D coordinates (x_{centre}, y_{centre}) and the depth value z_{center} to three-dimensional (3D) coordinates in the camera frame, the following steps are taken:

1. Obtaining Intrinsic Camera Parameters:
   The camera's intrinsic parameters include the focal lengths f_x and f_y, and the principal point coordinates (c_x, c_y). The following matrix K describes the camera's intrinsic parameters.

$$K = \begin{matrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{matrix} \qquad \text{(Equation 1)}$$

2. Calculating the 3D coordinates:
   The 3D coordinates (X, Y, Z) in the camera frame is calculated using the following formulas:

$$X = \left(\frac{(x_{centre} - c_x * z_{centre})}{f_x}\right) \qquad \text{(Equation 2)}$$

$$Y = \left(\frac{(y_{centre} - c_y * z_{centre})}{f_y}\right) \qquad \text{(Equation 3)}$$

$$Z = Z_{centre} \qquad \text{(Equation 4)}$$

Thus, the 3D coordinates (X, Y, Z) represent the position of the object in the camera frame, derived from the 2D image coordinates and the depth information.

2.3 Robot manipulation and control

The UArm Swift Pro Robot arm performs this use case's pick-and-place functionality. It has four degrees of freedom and can be used in various applications, such as laser engraving, 3D printing, and painting. The robot arm in this application is fitted with a suction gripper to pick up objects.

The ROS2 [9] tf2_ros package manages the transformation from the camera frame to the robot frame. The robot arm and the camera are configured in a Unified Description Format (URDF), which describes their physical properties and relative positions. Figure 2 shows the URDF visualisation for the camera and the robot arm.

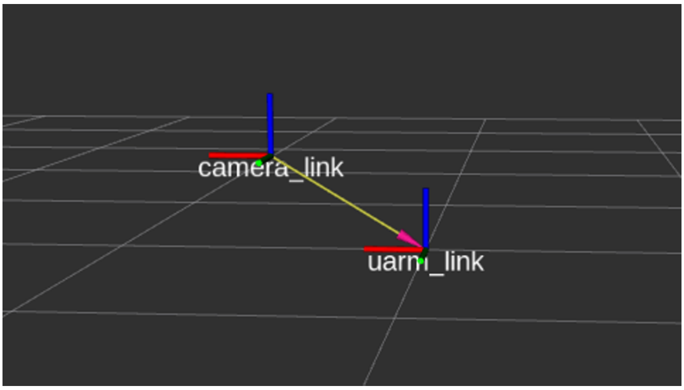

Fig. 2: URDF visualisation of the camera and robot arm position relationship.

The tf2_ros package is utilised to look up the transformation between the camera and the robot arm frames. The 3D coordinates of the object, calculated in the camera frame, are transformed into the robot arm frame using the transformation data provided in the tf2_ros package.

The UArm Python SDK is used for motion planning and control of the robot arm. A *home* position is configured to ensure the robotic arm returns to a known state after each action, enhancing repeatability and reliability. To achieve the functionality of sorting waste objects into different categories, two fixed locations are defined: one for paper waste and another for plastic waste.

3 Results

This results section evaluates waste detection and pick-and-place system performance using a ZED2i camera and the UArm Swift Pro robotic arm. The primary objectives are to assess the system's detection accuracy, localisation precision, and pick-and-place success rate.

3.1 Object detection

The key metrics used to measure the performance of object detection models are Precision, Recall, and Mean Average Precision (mAP) [10]. Considering these key metrics, Figure 2 and Figure 3 illustrate the performance of YOLOv5 and YOLOv8, respectively.

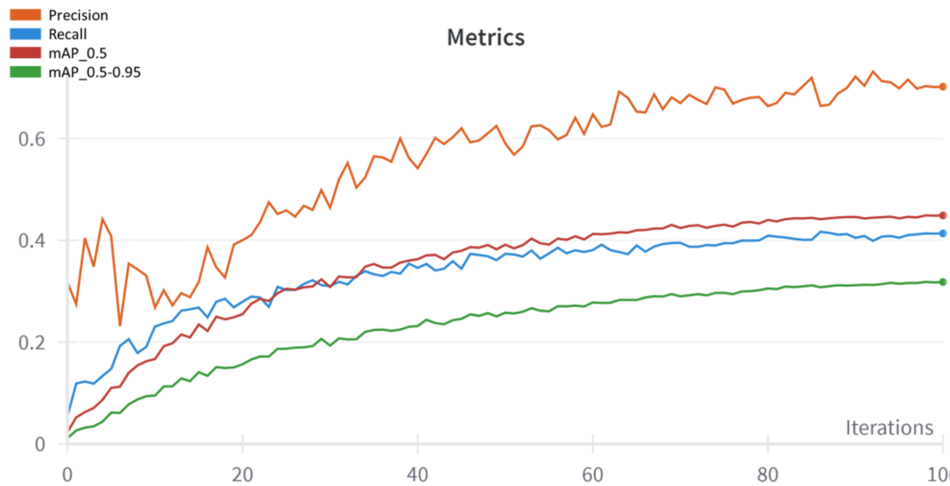

Fig. 3: Performance metrics for YOLOv5.

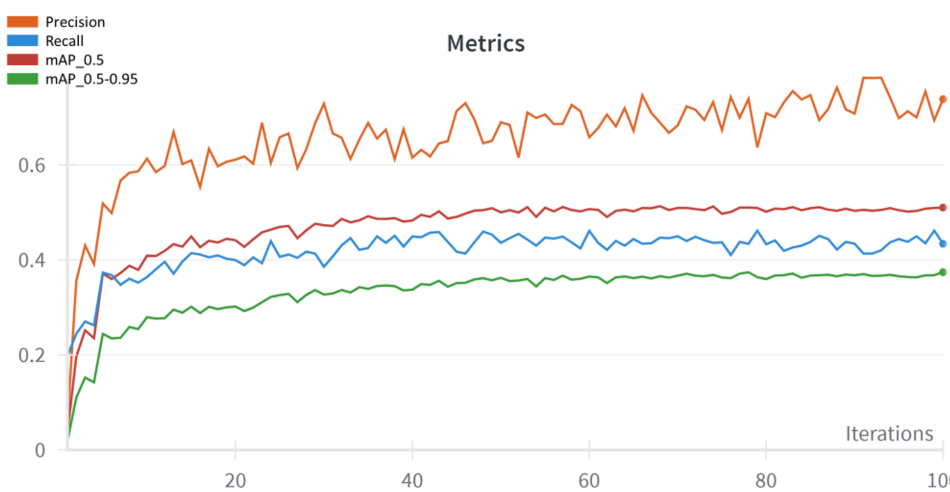

Fig. 4: Performance metrics for YOLOv8.

3.1.1 Precision

Precision metric measures how well the model distinguishes true objects from false positives, or the proportion of positive identifications which were correct.

$$Precision = \left(\frac{True\ Positives\ (TP)}{True\ Positives\ (TP) + False\ Positives\ (FP)}\right)$$

In the above graphs, precision starts with some volatility, common at the beginning of training as the model starts learning. The graph for YOLOv5 shows a more consistent upward trend, with precision ending higher at the final iterations. Whilst the graph for YOLOv8 shows an initial increase in precision, it levels off sooner and demonstrates more fluctuation in the later iterations, which indicates that the model is struggling to maintain consistent precision.

3.1.2 Recall

The Recall metric measures the model's ability to capture all relevant objects in an image, or the actual positive instances which were correctly identified.

$$Recall = \left(\frac{True\ Positives\ (TP)}{True\ Positives\ (TP) + False\ Negatives\ (FN)}\right)$$

In both YOLOv5 and YOLOv8, recall increases steadily before plateauing, indicating that the model is improving at finding all relevant classes. The recall graph for YOLOv8 is smoother, indicating a more stable learning process in identifying true positive cases over all the iterations.

3.1.3 Mean average precision (mAP)

There are two variants of mAP:

- **mAP_0.50:** Measure how well a model can locate objects with a moderate IoU (Intersection over Union) overlap of at least 50% with a ground truth object.
- **mAP_0.50-0.95:** Indicates that the mAP is computed by averaging the average precision (AP) over multiple IoU thresholds. More specifically, evaluates the model's performance at IoU thresholds of 0.5, 0.55, 0.6, …, up to 0.95.

In both YOLOv5 and YOLOv8, mAP_0.50 was improved. However, the graph for YOLOv8 shows a slight improvement to the graph for YOLOv5 toward the end of the training iterations, indicating that the model performs better at this IoU.
The curve for mAP_0.50-0.95 for YOLOv8 indicates that it performs better than YOLOv5. The graph also displays a less volatile increase, suggesting a smoother learning curve.

Overall, the data readings from the YOLOv8 graph suggest that it performs better than YOLOv5, particularly for Recall and mAP across both IoU thresholds.

3.2 Object position estimation

In this experiment, waste objects are categorised into two types: paper and plastic. A total of ten objects were used, with five objects in each category, as depicted in Figure 5. These objects were strategically placed within the camera's field of view for detection and position estimation.

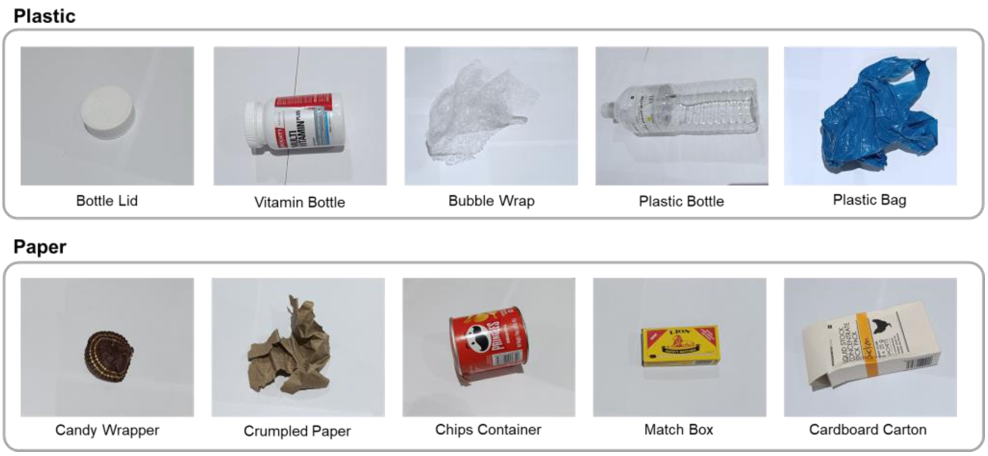

Fig. 5: Waste objects used in the experiments.

The YOLOv8 model performed better than the YOLOv5 during the model training stage, so it was selected for the detection stage. Once an object has been detected, its 3D position relative to the camera is calculated. The image provided in Fig. ***6*** illustrates the detected objects along with their calculated 3D coordinates.

For instance, analysing Figure 6a, the clear plastic bottle is detected at coordinates (-0.18, 0.07, 0.55), indicating its position to the left and slightly below the camera at a distance of 0.55 metres. Similarly, the small white bottle is detected at (0.23, 0.15, 0.61), and the box at (0.15, 0.22, 0.48). These coordinates accurately represent the objects' positions, which are essential for guiding the robot arm.

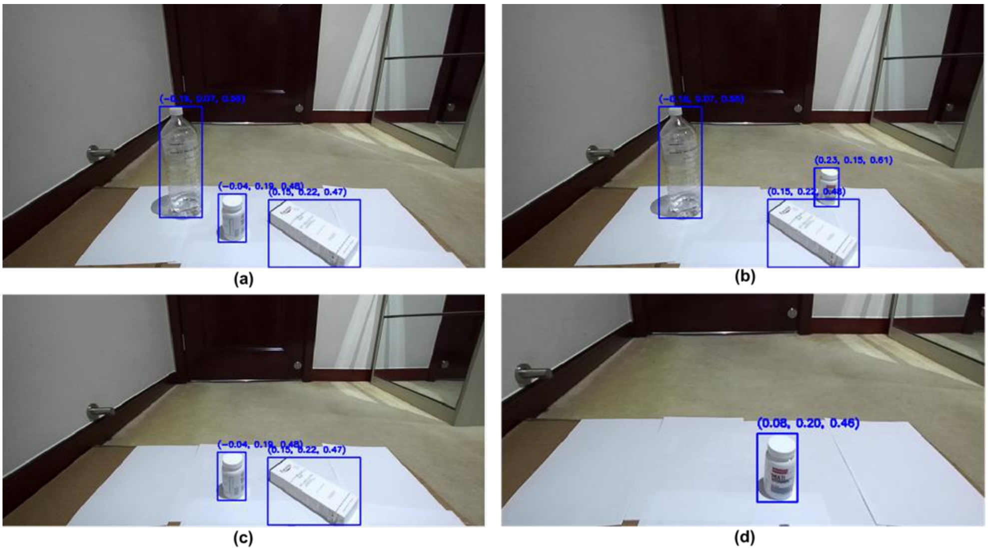

Fig. 6: Detected waste objects with 3D position estimations.

3.3 Robot arm manipulation

The transformation of the 3D coordinates from the camera frame to the robot arm frame is crucial for guiding the robot arm to pick up objects accurately. Using these transformed coordinates, the robot arm can move to the location of each object within its working space.

In practical application, the robot arm achieves a pick success rate of 70%. This is based on a series of tests involving 10 different waste samples, which was tested 10 times each from the same fixed location. However, the robot arm encounters significant challenges when picking up cylindrical objects, and successfully handling only 7 out of the 10 samples. Success (Pass) was defined as being able to pick up an object more than 50% of the time. Table 1 shows a detailed results table for each object pick success rate.

Table 1: Pick success rate for waste objects.

Object	**Category**	**Pick Count Success (Out of 10)**	**Pick Success (Pass >= 50%)**
Water Bottle	Plastic	3	0
Small Vitamin Bottle	Plastic	4	0
Cardboard Container	Paper	8	1
Candy Wrapper	Paper	9	1
Paper Cup	Paper	4	0
Plastic Bag	Plastic	10	1
Crumpled Paper	Paper	10	1
Small Box	Paper	8	1
Plastic Bottle Lid	Plastic	9	1
Bubble Wrap	Plastic	10	1

The primary issue lies in the designated "pick" coordinate, which is set at the centre of the object. The robot arm is currently limited by its inability to rotate its end-effector, making it difficult to grasp objects like bottles that require a specific orientation for successful pickup. Additionally, to the limitation of the degree of freedom with the end-effector, cylindrical objects have a curved surface, which can prevent the suction gripper from forming an airtight seal, which is necessary for a successful pick. This limitation highlights the need for advanced algorithms to compensate for its rotational constraints.

4 Conclusion

This paper presented a comprehensive system integrating YOLOv8 for object detection and a ZED2i depth camera for 3D position estimation, integrated with the UArm Swift Pro robotic arm to provide an end-to-end automated waste sorting system. Through experiments, waste objects categorised into paper and plastic were successfully detected and localised within the robot arm's operational environment. The performance of the YOLOv8 model was validated through consistent and accurate detection of various objects.

A critical component of the system is the transformation of 3D coordinates from the camera frame to the robot arm frame, enabling manipulation of detected objects. Despite achieving a pick success rate of 70%, challenges were identified, particularly in handling cylindrical objects like bottles.

Future work will focus on addressing the robot arm's rotational constraints and improving the robustness of the entire system. More statistical-based analysis will be conducted further to evaluate the performance of the object position estimation stage, and the overall efficiency of the system will be accessed by measuring the average time taken to complete each stage in the process.

Additionally, this system will be integrated with the Voyager mobile robotic platform [11]. This work will include modifying the platform to accommodate the UArm Swift Pro Robot Arm and the ZED2i depth camera. Mapping and autonomous navigation software will also be integrated with the platform to achieve a fully autonomous waste collection robot capable of exploring an area and collecting waste from that area.

In conclusion, this study showcases a viable approach to leveraging deep learning and robotic integration for intelligent waste management, paving the way for more sophisticated and automated solutions in the field.

References

[1] Polasi Tumisang, Matinise Sihle, and Oelofse Suzan, "South African Municipal Waste Management Systems: Challenges and Solutions | International Environmental Technology Centre." Accessed: Jul. 04, 2024. [Online]. Available: https://www.unep.org/ietc/resources/publication/south-african-municipal-waste-management-systems-challenges-and-solutions

[2] L. K. Haywood, T. Kapwata, S. Oelofse, G. Breetzke, and C. Y. Wright, "Waste Disposal Practices in Low-Income Settlements of South Africa," *International Journal of Environmental Research and Public Health 2021, Vol. 18, Page 8176*, vol. 18, no. 15, p. 8176, Aug. 2021, doi: 10.3390/IJERPH18158176.

[3] L. K. Haywood, T. Kapwata, S. Oelofse, G. Breetzke, and C. Y. Wright, “Waste Disposal Practices in Low-Income Settlements of South Africa,” *International Journal of Environmental Research and Public Health 2021, Vol. 18, Page 8176*, vol. 18, no. 15, p. 8176, Aug. 2021, doi: 10.3390/IJERPH18158176.

[4] T.-A. T. Luu, H.-M. Le, M.-Q. Vu, B.-V. Nguyen, and (Văn, “AI application for solid waste sorting in Global South”.

[5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-December, pp. 779–788, Jun. 2015, doi: 10.1109/CVPR.2016.91.

[6] P. F. Proença and P. Simões, “TACO: Trash Annotations in Context for Litter Detection,” Mar. 2020, Accessed: Jul. 04, 2024. [Online]. Available: https://arxiv.org/abs/2003.06975v2

[7] Glenn Jocher, “YOLOv5.” Accessed: Jul. 04, 2024. [Online]. Available: https://github.com/ultralytics/yolov5

[8] Glenn Jocher, “YOLOv8.” Accessed: Jul. 04, 2024. [Online]. Available: https://github.com/ultralytics/ultralytics

[9] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, “Robot Operating System 2: Design, Architecture, and Uses In The Wild,” *Sci Robot*, vol. 7, no. 66, Nov. 2022, doi: 10.1126/scirobotics.abm6074.

[10] R. Padilla, W. L. Passos, T. L. B. Dias, S. L. Netto, and E. A. B. Da Silva, “A Comparative Analysis of Object Detection Metrics with a Companion Open-Source Toolkit,” *Electronics 2021, Vol. 10, Page 279*, vol. 10, no. 3, p. 279, Jan. 2021, doi: 10.3390/ELECTRONICS10030279.

[11] K. Purdon, J. S. Dickens, W. De Ronde, K. Ramruthan, and G. Crafford, “Voyager, a ground mobile robotic platform for research development,” *RAPDASA-RobMech-PRASA-AMI Conference, CSIR International Convention Centre, Pretoria, South Africa, 30 October – 2 November 2023*, 2023, doi: 10.1051/matecconf/202338804016.

[12] Dwyer, B., Nelson, J., Hansen, T., et. al. (2024). Roboflow (Version 1.0) [Software].