

SORDI.ai HACKATHON 2022

INDUSTRIAL AI OBJECT DETECTION CHALLENGE

sordi.ai/hackathon



**BMW
GROUP**



NVIDIA

Microsoft

idealworks

inmind.ai

HW

OBJECTIVE⁺
IMAGE RECOGNITION STUDIO

OpenACC

BMW GROUP + QUT
DESIGN ACADEMY

Contents

| | | |
|----------|--|-----------|
| 1 | Introduction | 1 |
| 2 | Problem Statement | 1 |
| 2.1 | Assets Description | 2 |
| 2.2 | List of Assumptions | 3 |
| 3 | Provided Dataset | 4 |
| 3.1 | SORDI.ai Hackathon Filesystem | 4 |
| 3.2 | BMW JSON Labeling Format | 6 |
| 3.3 | Labels Objectclasses.json File | 6 |
| 4 | Hackathon Approach | 7 |
| 4.1 | Training Objective | 7 |
| 4.2 | Training Evaluation | 7 |
| 5 | Scoring & Submission | 8 |
| 5.1 | Numerical Evaluation | 8 |
| 5.2 | Bonus Points | 9 |
| 6 | Hints | 10 |

1 Introduction

In today’s world, Machine Learning (ML) and Computer Vision (CV) have gained unprecedented interest, with extensive research in academia and increasing implementation and adoption in the industry. Object detection, an essential subset of CV applications, has been widely integrated into various industrial domains and applications to leverage robots’ decision-making, therefore automating and enhancing production: Object detection provides computers with the ability to perceive the world visually, and mimics human vision to detect and identify salient objects in an image or a video, allowing computers to react accordingly based on dedicated algorithms.

One major and recurring challenge is the need to train and deploy different object detection models for specific industrial use cases. This in turn requires a significant amount of high-quality training data which is oftentimes unavailable. Many challenges are faced when collecting a dataset, especially in industrial domains due to confidentiality, safety, plant setup, and many more considerations. To overcome this issue, BMW Group has launched in collaboration with its partners NVIDIA, Microsoft, idealworks, inmind.ai, and many prestigious universities from around the globe, the world’s most comprehensive and extensive Synthetic Object Recognition Dataset for Industries – SORDI! [1, 2]

SORDI.ai contains hundreds of thousands of photo-realistic images for 80+ distinct object classes highlighting logistics, transportation, signage, tools, and office assets. They are path-trace rendered in NVIDIA Omniverse using the latest RTX technology and Pixar’s USD pipeline [3]. SORDI.ai significantly simplifies training CV models to recognize equipment in numerous factory environments. Additionally, to its practical benefit, the dataset simultaneously provides a new global benchmark for research and science in the usage of synthetic data.

2 Problem Statement

This contest aims at utilizing and popularizing the SORDI.ai dataset for industrial object detection and recognition. It consists of two subsequent stages. In Stage 1, each participating team is asked to train an object detection model based on a provided subset from SORDI.ai, consisting of a select number of 17 assets with their assigned labels shown in Figure 1. Teams will compete to maximize object detection quality. They have the freedom to choose any machine learning or deep learning approach without any constraints on model type, framework, or technologies. Nonetheless, teams are encouraged to use BMW TechOffice tools published on GitHub. The top-10 ranking teams will be shortlisted to participate in Stage 2 of the Contest, where they will continue the development of their solution with additional requirements and constraints provided by the organizers. Further details will be communicated to participants who make it to Stage 2.



Figure 1: 1st stage hackathon’s 17 industrial assets (from left to right, top to bottom): stillage_close, stillage_open, l_klt_6147, l_klt_8210, l_klt_4147, pallet, jack, forklift, str, bicycle, dolly, exit_sign, fire_extinguisher, spring_post, locker, cabinet, and cardboard_box

2.1 Assets Description

Please find below, a brief description and explanation of the 17 assets:

- **Stillage:** It is similar to a pallet or skid, but it has a cage, sides, or another type of support that is specially fitted to the substance it is designed to carry, and transport. Some are intended to be stacked. We consider both closed and opened cage of the stillage: **stillage_close** (ID = 1002), and **stillage_open** (ID = 1003) respectively.
- **Small load carrier box:** aka. KLT box (from German: Kleinladungsträger). It is an industrial stacking container. We consider three different dimensions of the large KLT box: 6147 ($60 \times 40 \times 14,7$ cm), 8210 ($80 \times 60 \times 22$ cm), and 4147 ($40 \times 30 \times 14,7$ cm) corresponding to **l_klt_6147** (ID = 1012), **l_klt_8210** (ID = 1013), **l_klt_4147** (ID = 1011) respectively.
- **Pallet:** A **pallet** (ID = 1100) is a flat transport structure that holds products securely while being hoisted by a forklift, pallet jack, a front loader, a jacking device, or an erect crane. A pallet is the structural basis of a unit load, allowing for efficient handling and storage.
- **Pallet jack:** A **jack** (ID = 1120) is the most basic type of forklift and is used to lift and transport pallets inside an industrial setting.
- **Forklift:** A **forklift** (ID = 2010) is a motorized industrial truck used for short-distance lifting and moving of items.

-
- **Smart transport robot:** aka. **str** (ID = 2050) is an innovative driverless logistic transport system that transports items inside factory environments without using floor markers. For more information, please check: <https://www.idealworks.de>.
 - **Bicylce:** A **bicycle** (ID = 2000) is a pedal-driven, human-powered or motor-powered aided vehicle with two wheels mounted to a frame, one behind the other.
 - **Pallet dolly:** The **dolly** (ID = 1110) is a 4-wheel system designed to simplify the storage and movement of specific dimension pallets.
 - **Exit sign:** It is a pictogram displayed in a building that indicates the location of the nearest emergency exit in case of immediate evacuation. The **exit_sign** (ID = 4000) is intended to be recognizable and intelligible to anybody.
 - **Fire extinguisher:** The **fire_extinguisher** (ID = 5010) is an active fire protection equipment that is intended to extinguish or control minor flames, which are frequently utilized in emergencies.
 - **Spring post:** It is bolted to the ground to guide traffic or to warn of impending risks. The **spring_post** (ID = 1135) may be used in any circumstance where traffic must be separated quickly and safely.
 - **Locker:** A **locker** (ID = 1030) is a set of 4×3 small lockable cupboards, placed together for public safe storage use.
 - **Cabinet:** A **cabinet** (ID = 1040) is a lockable, and functional goods carrier.
 - **Cardboard box:** A **cardboard_box** (ID = 1070) is a prefabricated lightweight industrial box that is primarily used for packing goods and supplies.

2.2 List of Assumptions

Towards solving this use case, we recommend you pay attention to the following assumptions:

- Multiple 3D models are associated with the same class, such as bicycles, fire extinguishers, etc.
- Texture randomization is considered for some other 3D models, such as pallet jacks.
- Assets may not be presented as singletons. However, they can be combined together.
- All assets may be presented at different scales.

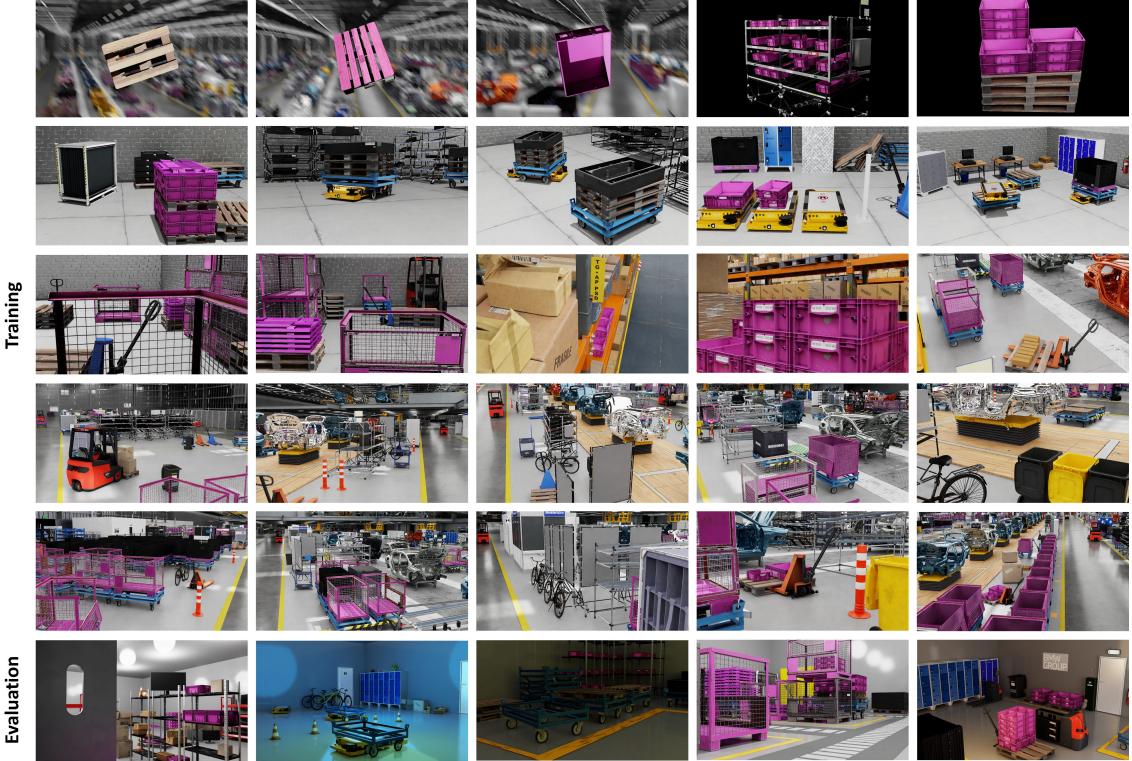


Figure 2: Hackathon training and evaluation datasets’ sample

3 Provided Dataset

The provided dataset is a subset of SORDI.ai approx. consisting of approx. 200K rendered images divided in 25 folders of single-asset images, and multiple-asset images in a warehouse, industrial areas, and BMW’s Regensburg digital twin areas (check Figure 2): 194,337 images are labeled with bounding boxes (bbox). All training bboxes are generated with Isaac Sim, which is highly accurate. Therefore, it is possible to encounter labels for extremely small regions (e.g., zero area bounding boxes) that we recommend cleaning before feeding them to your model. Moreover, all labels are encoded in BMW JSON format (check Section 3.2). While the training datasets are considered for training the hackathon’s models, 217 unlabeled images in “SORDI_2022_h4021_plant region (eval)” folder are **exclusively reserved for testing only**. This hackathon dataset is structured as shown in Figure 3.

3.1 SORDI.ai Hackathon Filesystem

The SORDI.ai dataset contains multiple sub-datasets, each is structured identically:

For more information about the SORDI.ai dataset and our open-source projects,
please visit sordi.ai and our GitHub: github.com/BMW-InnovationLab

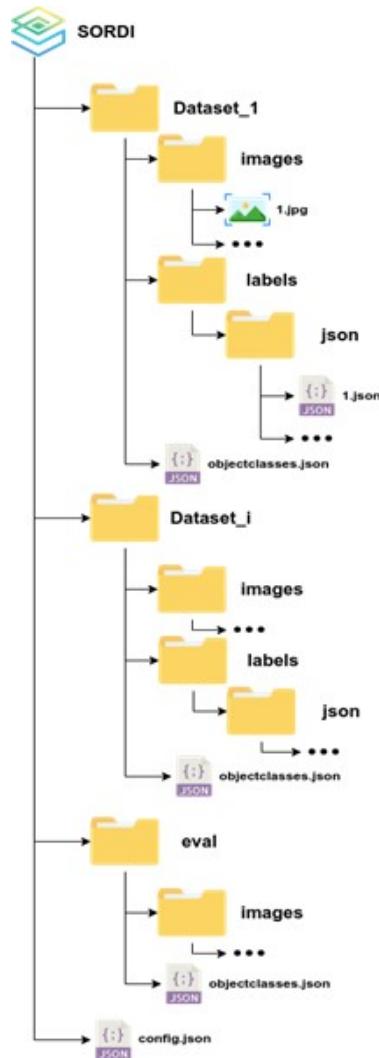


Figure 3: Hackathon dataset filesystem

1. Images folder: containing all the training images in JPEG format.
2. Labels folder with JSON sub-folder: it contains the respective image label in BMW JSON format.
3. Objectclasses.json contains the training object classes with their relative IDs.

For reference, please check the online existing Label Tool: <https://github.com/BMW-InnovationLab/BMW-Labeltool-Lite>.

Additionally, we omitted the label folder from the “eval” dataset. It is used for model evaluation. All the folders metadata can be found in the “config.json” file.

3.2 BMW JSON Labeling Format

Each label file consists of an array of elements where each element represents a bounding box (bbox) with its respective class. Taken all together, it is defined as follow:

```
1 [ {  
2     "Id": <-- int: label id -->,  
3     "ObjectName": <-- string: object class name -->,  
4     "ObjectClassId": <-- int: object class id as mentioned in  
5         the Objectclasses.json -->,  
6     "Left": <-- int: left bbox coordinates -->,  
7     "Top": <-- int: top bbox coordinates -->,  
8     "Right": <-- int: right bbox coordinates -->,  
9     "Bottom": <-- int: bottom bbox coordinates -->  
10 },  
11 {  
12     "Id": 18111997,  
13     "ObjectName": "example",  
14     "ObjectClassId": 7,  
15     "Left": 294,  
16     "Top": 115,  
17     "Right": 314,  
18     "Bottom": 154  
19 } ]
```

3.3 Labels Objectclasses.json File

All training classes are stored in the Objectclasses.json that have the following format:

```
1 [ {  
2     "Id": <-- int: object class id -->,  
3     "Name": <-- string: object class name -->,  
4     "ColorCode": <-- string: bbox color, when visualized -->,  
5     "ParentObjectClassID": <-- int: parent object class id -->  
6 },  
7 {  
8     "Id": 7,  
9     "ObjectName": "example",
```

```
10 "ColorCode": "ff005b",
11 "ParentObjectClassID": null
12 }]
```

4 Hackathon Approach

In this section, we will explain the main criteria for training and evaluating the models.

4.1 Training Objective

The objective of stage 1 of the hackathon is to train an object detection model to achieve the highest accuracy possible on the evaluation dataset. There is no limitation on the architecture or platform used to train the model.

The users are free to use any existing tools to train the model, as well as using the public tools on the following GitHub: <https://github.com/BMW-InnovationLab/BMW-TensorFlow-Training-GUI>.

4.2 Training Evaluation

The evaluation process will take place using a custom-made evaluation algorithm based on the Pascal VOC dataset evaluation process mentioned in [4].

The algorithm will evaluate based on Intersection over Union (IoU) score and detection confidence, while the scoring will take into consideration:

1. **mAP (mean Average Precision)** for training classes using all point interpolation method:

$$mAP = \frac{1}{n} \sum_{k=1}^{k=n} AP_k \quad (1)$$

where AP_k is the AP of a class k , and n is the number of all classes.

2. **Accuracy** measuring the correct detections based on all detections:

$$Acc = \frac{\text{Correct Answers}}{\text{All Possibilities}} = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

The IoU threshold used for the model evaluation is set to the default threshold of **IoU@0.5** and a prediction confidence threshold equal to 0.5.

5 Scoring & Submission

The scoring of each submission will be based on the best mAP and accuracy scores, which are automatically calculated based on the submitted evaluation set in addition to granted bonus points.

5.1 Numerical Evaluation

Every team of the 1st stage is required to submit an evaluation CSV “eval.csv” file on the hackathon portal profile. This file will comprise your model inference on the provided evaluation dataset. The required CSV structure contains 11 columns:

| detection_id | image_name | image_width | image_height | object_class_id | object_class_name | bbox_left | bbox_top | bbox_right | bbox_bottom | confidence |
|--------------|------------|-------------|--------------|-----------------|-------------------|-----------|----------|------------|-------------|------------|
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |

Table 1: eval.csv tabular sample

- Detection ID: “**detection_id**”, a unique integer related to every prediction-made
- Image Information: consists of 3 values, referring to the image settings used for the prediction. Kindly note that an image can be associated with multiple predictions.
 - “**image_name**”: it is a string mentioning the image name inside the evaluation test folder including its file extension
 - “**image_width**”: it is the image width in integer and pixel
 - “**image_height**”: it is the image height in integer and pixel
- Prediction: it is the prediction made on the image mentioned in the previous columns. An object detection consists of (1) recognizing and (2) localizing a specific asset. Therefore, we expect 2 predictions:
 - Object Recognition:
 - * “**object_class_id**” is an integer referring to the recognized object ID out of the total 17 assets of this challenge.
 - * “**object_class_name**” is a lower capital string referring to the recognized object name.

Please, refer to the objectclasses.json documents for the correct association and format of the object class ID and name, or the section [2.1](#) above.

-
- Object Detection: 4 integers in pixel consisting of the left, top, right and bottom coordinates of the prediction’s bbox: “`bbox_left`”, “`bbox_top`”, “`bbox_right`”, and “`bbox_bottom`” respectively.
 - Confidence: The **confidence** is a float number marking how much the model is certain about this row’s prediction it has made. The confidence score should be a percentage between 0 and 100.

Taken all together, the eval.csv file has 11 comma-separated values. The 1st row is:
`detection_id`, `image_name`, `image_width`, `image_height`, `object_class_id`,
`object_class_name`, `bbox_left`, `bbox_top`, `bbox_right`, `bbox_bottom`, `confidence`

5.2 Bonus Points

Bonus points will be given to the teams who star and hopefully use the three below public BMW TechOffice tools. They are shared on GitHub: <https://www.github.com/BMW-InnovationLab>:

- We launched NoCode AI tools to facilitate object detection training with little to no-code based on TensorFlow 2 <https://github.com/BMW-InnovationLab/BMW-TensorFlow-Training-GUI>, and an evaluation tool to assess trained models: <https://github.com/BMW-InnovationLab/SORDI-AI-Evaluation-GUI>.
- Additionally, users, who want to program on their own, are encouraged to use the SORDI data pipeline reader: <https://github.com/BMW-InnovationLab/SORDI-Data-Pipeline-Reader>, for the ease of data manipulation and processing.

Therefore, taken all together, the submission of each team **must** include:

1. **eval.csv** file: containing the detection based on the provided evaluation dataset as described in the previous section.
2. **Participant’s GitHub Username**: to cross-check if a bonus point is earned based on using and starring BMW TechOffice tools. A “`github_users.csv`” must be submitted. It consists of one column. The first row is `username`. Every row has a distinct GitHub username that has starred the above-mentioned GitHub repositories.
3. **PDF document** (For the Top-10 teams **ONLY**): a maximum 2-page long report detailing the implementation process and the method used, e.g., model, architecture, framework, training steps, hardware, libraries, experiments, etc.

6 Hints

During the hackathon, we recommend you stay updated through our official SORDI.ai website for the latest news. For additional hints related to the challenge, please follow Marc Kamradt (<https://www.linkedin.com/in/marc-kamradt>) on LinkedIn.

Good Luck!

References

- [1] BMW Group. BMW Group publishes SORDI, the largest open-source dataset by far for super-efficient AI applications in production. <https://www.press.bmwgroup.com/global/article/detail/T0375993EN/bmw-group-publishes-sordi-the-largest-open-source-dataset-by-far-for-super-efficient-ai-applications-in-production?language=en>. Online; accessed 23 March 2022.
- [2] Marc Kamradt, Jimmy Nassif, Chafic Abou Akar, Daniel Jess, Dylan Sheppard, Roger Limberger, Sebastian Heim. Creating the Future: Creating the World’s Largest Synthetic Object Recognition Dataset for Industry (SORDI). <https://www.nvidia.com/en-us/on-demand/session/gtcspring22-s41142/?playlistId=playList-1f1c32ea-247f-4959-864f-78703b4c78de>. Online; accessed 16 September 2022.
- [3] Chafic Abou Akar, Jimmy Tekli, Daniel Jess, Mario Khouri, Marc Kamradt, and Michael Guthe. Synthetic object recognition dataset for industries. In *Proceedings... Conference on Graphics, Patterns and Images*, 35. (SIBGRAPI), 2022.
- [4] PASCAL. The PASCAL Visual Object Classes Homepage. <http://host.robots.ox.ac.uk/pascal/VOC/>. Online; accessed 16 September 2022.