Background

Quality management and control constitute a crucial part in automation and digital transformation of factories. In previous years, Edison Cars AG performed quality control manually, as employees inspected and evaluated the car parts. So, their quality control only was based on the knowledge of their experienced employees. Following the switch to production of electric cars and a shortage of skilled employees, the Edison Cars AG aims to automate quality control and management. To achieve the best results, they plan to introduce a computer vision system in their final production step due to recurring damage caused by scratches after mounting car parts on the back side. The system should detect whether the backside of a car was produced correctly or if there is a scratch on the backside of the car. Based on the detection, the system should be capable of determining the coordinates of the scratch and recommend ways of dealing with it. Further, the system should be expandable to more features, e.g., a scratch length measurement feature.

Datasets

For a good object detection approach, it is very important to choose the right dataset. To develop the new quality control system of the Edison Cars AG, you are provided with four datasets. At first the Edison Cars AG only produced blue electric cars. Because in the future the production should be expanded to also manufacture white and red cars the system should also work for that kind of car colors. You find the datasets in the folder “Data”. Each dataset consists of two folders “scratch” and “no_scratch”. In the “scratch” folder every car has a scratch on the backside and in the “no_scratch” folder none of the cars has a scratch on it.

Task

Choose **one** of the four above-mentioned datasets and label the data with a labelling strategy of your choice. Split the data in a train and validation/test set. Implement an object detection approach, which can detect a scratch on the backside of the car. The approach should be expandable for further tasks.

Input

For this task, you will be provided with the following:

- Four different datasets, which are described above
- Two possible instructions for the labeling of the data
- Two different textual instructions for a possible solution. You can also create a solution on your own in a common programming language. However, make sure you use the correct output format (see also Submission)
- Further information for object detection and machine learning: These are possible approaches that have different advantages and disadvantages. Of course, you can also use other approaches and resources:
    - Code quality: browserstack.com
    - Object detection: https://www.tensorflow.org/hub/tutorials/object_detection, https://de.mathworks.com/discovery/object-detection.html
    - Bounding Boxes and object detection: https://towardsdatascience.com/object-detection-with-neural-networks-a4e2c46b4491

Submission

The following documents must be emailed to s3g@fim-rc.de as one zip folder by 09:00 on 16.05.2023:

- Code file(s) to reproduce the results. Should the setup of a special environment or packages be necessary, a step-by-step guide is also needed

- Please export your model as described here: https://tf.wiki/en/deployment/export.html. Please send the whole "export" folder to us. Also make sure that your model is in ".pb" format.
- Please let us know about the installed versions of your modules and libraries. Using "pip list" might help you with that. Without the information about installed versions, loading and evaluating your model will hardly be possible.
- Your model will be tested with production data. The performance of your model will lead to your results.
- A PowerPoint presentation explaining your approach. You should be able to present your approach in the next lecture.

Keep in mind

The content provided here serves only as a starting point. Feel free to use your own approaches and algorithms to get the best possible object detection.

Use only the datasets provided.

The following aspects are important for the assessment of your submission:

- Code quality: The code must be executable, readable, commented, and adhere to the output format
- Detection Accuracy: The accuracy will be proved on a hidden test set.
- Explanation of the approach should be understandable