Duferco: Project Proposal Neural Wave Hackathon 2024

Reference:
Fabio Iannello
fabio.iannello@duferco.com
+41793129008

Company Overview:

Duferco is a global company with interests across various industries, including energy retail and trading, steel manufacturing, insurance, and logistics. With a distributed workforce and a strong presence in Europe and the Americas, Duferco has positioned technology and data as core components of its strategy. Over the past few years, we have embraced technological innovation across all business units, using data and artificial intelligence (AI) to drive decision-making processes.

Project Introduction & Motivation:

In our steel manufacturing division, we are focused on integrating machine learning (ML) and AI solutions to optimize our production processes. One of the key areas for innovation is within our rolling mill operations. Our goal is to support plant operators by leveraging AI to improve decision-making, particularly through computer vision technologies. By automating routine tasks, we aim to reduce operational complexity, minimize human error, and enhance overall production efficiency.

Our challenge for the hackathon participants is to develop a solution that uses AI to assist in real-time decision-making within the rolling mill. Specifically, participants will focus on detecting and assessing the interaction between steel bars and the mechanical stopper, which is a crucial part of the production process.

Problem Statement:

The task is to build a model that can detect the alignment of steel bars with a stopper wall using image data. The alignment process is crucial in the rolling mill as it directly affects the cutting process of the steel bars. Operators currently verify this alignment through visual inspection of live video feeds. The automation of this task through AI can reduce operator workload and increase efficiency.

Objective:

The model needs to detect whether all the steel bars are touching the stopper simultaneously (i.e., they are aligned). If they are not aligned, the cutting process should not proceed.

Key Considerations:

- The model must classify each image into one of two classes: `ALIGNED` (all bars touching the stopper) or `NOT_ALIGNED` (at least one bar not fully touching the stopper).
- There are two machines (Stopper 1 and Stopper 2) with slight variations in camera perspective and lighting conditions, but the model must generalize and perform equally well on both machines.
- The evaluation metric will be the F-beta score on the test dataset (see below for details).
- The classification shall work in real time. Consider an inference time of less than 0.5 seconds/image as a reasonable target on commercial laptop hardware, even without GPU (see section below for additional details).

Note that, there are no sensors for detecting the contact between bars and stopper, and the evaluation is performed by visually observing the same cameras that you are given in the dataset. Some relevant examples will be provided of scenarios in which bars are considered aligned or not, so that participants can learn how to properly identify situations. We will try to reduce or avoid completely the use of ambiguous images, in which alignment or not is hard to identify event from an expert plant operator.

Data Description:

Participants will have access to thousands of labeled images containing various scenarios, including:

- **Background Only**: No steel bars or stopper in view.
- **Bars Only:** Steel bars visible, but no stopper.
- **Stopper Only:** Stopper visible, but no steel bars.
- **Bars and Stopper:** Both visible, but the alignment may vary (fully aligned, partially aligned, or not aligned).

The images will come from two different machines (Stopper 1 and Stopper 2), and the test set, which participants will be evaluated on, will include images from both machines.

Potential Solution:

Participants are expected to build a solution that takes an image as input and returns the (binary) classification:

- **0 for NOT_ALIGNED**

- **1 for ALIGNED**

Several approaches can be taken, such as:

- **Sequential Detection:** First detect the presence of bars and the stopper, followed by determining their alignment.
- **Classification, Object Detection or Segmentation:** Develop a strategy using classification, object detection or segmentation techniques to differentiate between the various scenarios and detect alignment.

Participants should use open-source libraries such as PyTorch, TensorFlow, or OpenCV, and the solution must be implemented in Python. The code should utilize models that are freely available or that participants can train or fine-tune from scratch.

Performance metric:

The problem essentially boils down to a binary classification problem to detect whether bars are aligned with the stopper or not.

The metric of choice to evaluate the model is the F-beta score:

$$F_\beta = (1+\beta^2) \times \frac{\text{Precision} \times \text{Recall}}{(\beta^2 \times \text{Precision}) + \text{Recall}}$$

with $\beta = 0.5$, and:

$$\text{Precision} = \frac{TP}{TP+FP} \qquad \text{Recall} = \frac{TP}{TP+FN}$$

with:

- TP (True Positives): The number of positive instances correctly classified.
- TN (True Negatives): The number of negative instances correctly classified.
- FP (False Positives): The number of negative instances incorrectly classified as positive.
- FN (False Negatives): The number of positive instances incorrectly classified as negative.

Where positive instance indicates bars ALIGNED.

Computational requirements:

The proposed solution should process images in real time for use by operators. As a reference, an inference time of 0.5 seconds per image is a reasonable target on standard commercial hardware (even without a GPU). Since we do not know the specific hardware that will be used to evaluate the proposed solutions, we are not imposing strict constraints on inference time for ranking purposes. However, we highly value participants' efforts to develop solutions that can support real-time inference.

Additional Information:

The use of proprietary or commercial software is prohibited. The participants must rely on open-source tools. They can access the following resources:

- **Dataset:** Labeled images for training and validation.
- **Libraries**: PyTorch, TensorFlow, OpenCV, or similar

Sample images

In this section, we present several example images to clarify the objective of the classification task. These images represent the only method operators in the plant have to assess the alignment of bars. The assessment relies entirely on visual inspection, where operators determine whether the bars are aligned based on the images.

It is important to note that in some cases, determining alignment may be challenging due to uncertainties introduced by factors such as perspective, vibration, shadows, and lighting conditions. Recognizing the complexity of making precise judgments in these situations, we have provided a set of pre-classified images to help participants better understand the identification process. This guidance is intended to assist with the labeling of the dataset, regardless of the technique participants choose to apply.

Figure 1 This picture shows two steel bars that are moving towards a stopper on a rolling table.

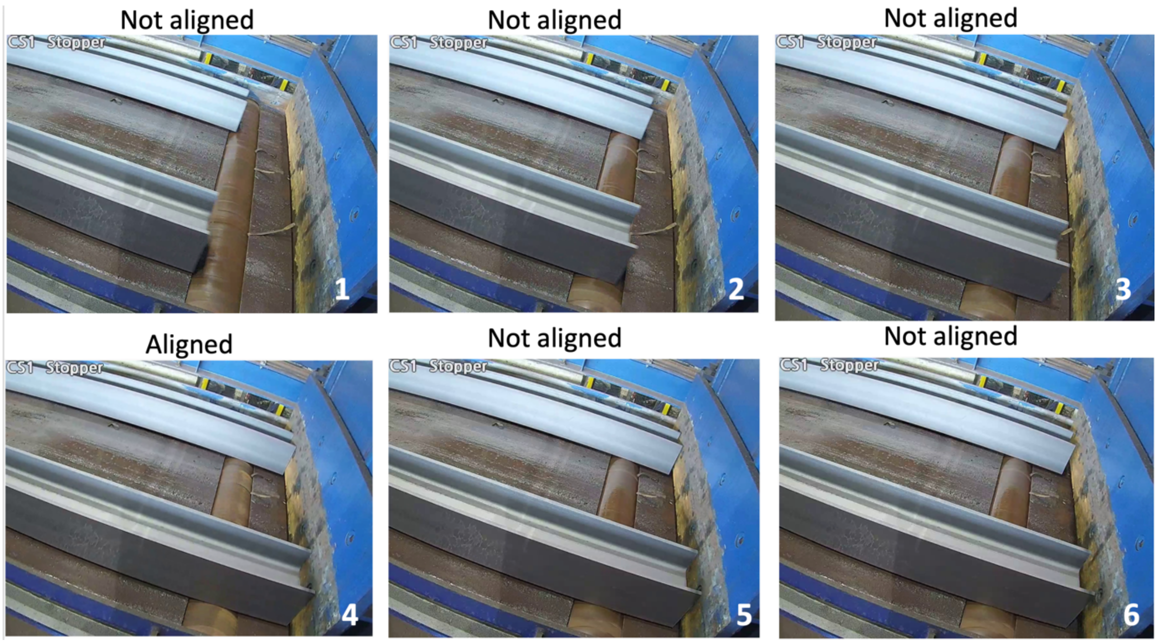

Figure 2 Sequence of bars approaching the stopper. All pictures show non-aligned bars, except example sub fig. 4. For example, sub fig. 5 show bars very close to the stopper, but not touching (compare with sub fig. 4)

Figure 3 Bars aligned (left) vs bars not aligned (right). The bars on the right figure are close to the stopper, but they are not in contact.

Figure 4 Example with missing stopper. Here bars are not aligned

Figure 5 Scenario with only stopper and no bars

Figure 6 Examples of alignment

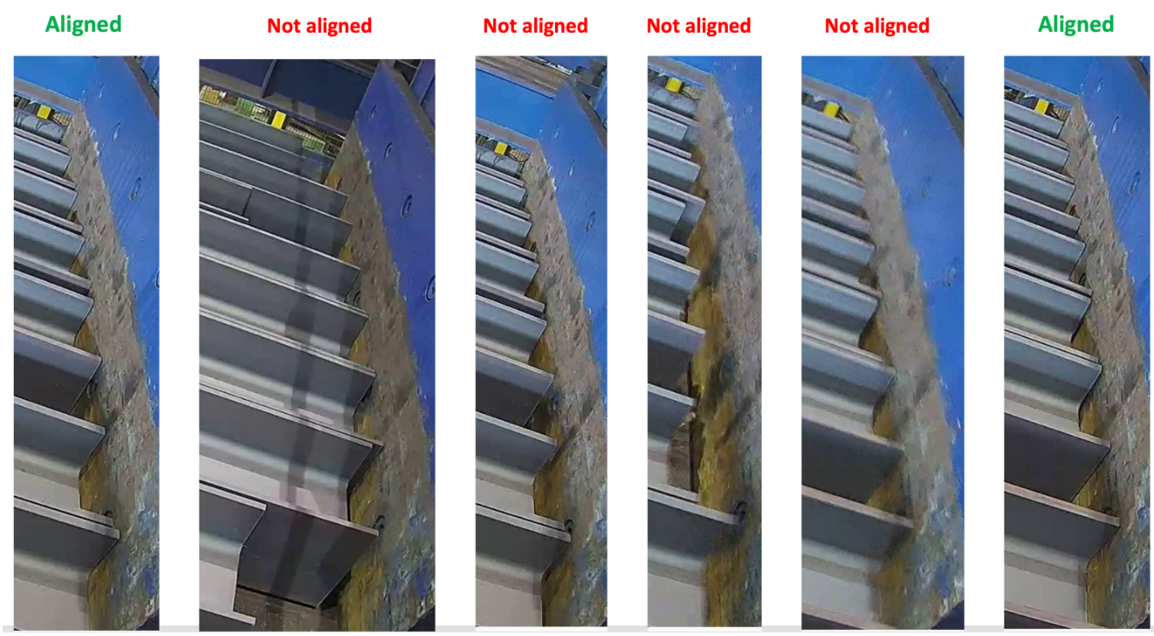

Figure 7 Other examples of alignment