Neural Wave Hackathon 2024 Report

CUDA_Libre Team: Arslan Ali, Daniel Sima, Kevin Ortiz

October 27, 2024

Project Title:

Duferco Steel Bar Alignment Detection

Objective

The objective of this project is to automate the verification of steel bar alignment in the rolling mill, a task currently performed manually by plant operators. At present, operators visually inspect the alignment of steel bars before initiating the cutting process. This manual inspection can be time-consuming and may lead to decreased focus over time, increasing the risk of human error.

By automating this alignment verification using AI, our solution aims to streamline the process, reducing operational complexity and enhancing efficiency. This will allow plant operators to focus on other critical tasks, minimize errors, and improve the overall productivity of the cutting process.

Approach

The primary challenge of this project was the lack of labeled data within an extensive dataset. We were provided with 15,630 images; however, only 296 of these were labeled with alignment information, where 41 images were classified as **aligned** and the remaining 255 as **not aligned**.

Our solution consists of two main stages: **Labeling** and **Classification**. In the labeling phase, we focused on efficiently generating accurate labels for the majority of the dataset. In the classification phase, we utilized these labels to train a model capable of predicting alignment with high accuracy.

Labeling Strategy

Given the large volume of unlabeled data, manually labeling the entire dataset was resource intesive and time consuming. However, a substantial number of labeled examples were necessary to train a robust classification model effectively. Therefore, we devised a semi-supervised labeling strategy to expand the labeled dataset without manually annotating each image, combining **manual labeling** with **pseudo-labeling**.

First, we manually labeled a subset of 5,000 images from the dataset to establish a reliable foundation of labeled data. To label the remaining images, we employed a pseudo-labeling approach based on **image similarity**. Specifically, we used **DINO v2** [1], a self-supervised vision transformer, to generate high-dimensional embeddings for each image. These embeddings capture semantic information about the images, allowing us to measure similarity between images in an embedding space.

We then applied a **K-Nearest Neighbors (KNN)** algorithm to find similar images for each unlabeled sample in the embedding space. By leveraging **majority voting** among the K-nearest neighbors, we assigned alignment labels to the unlabeled images based on their closest labeled neighbors. This method provided an efficient way to generate reliable labels while reducing the need for extensive manual labeling. To ensure the effectiveness of this pseudo-labeling technique, we evaluated its performance using a separate test set not included in the embedding space.

Classifier Training

With an expanded labeled dataset, we proceeded to the classification phase. Our model of choice was **EfficientNet B0** [2], a convolutional neural network architecture known for achieving high accuracy while maintaining computational efficiency. EfficientNet B0 is part of the EfficientNet family, which employs compound scaling to optimize model depth, width, and resolution, enabling the architecture to

handle diverse visual datasets efficiently. This scalability makes it well-suited for our application, where balancing speed and accuracy is essential.

Initially, we experimented with **MobileNetV2** due to its lightweight architecture and efficiency in mobile and embedded applications. However, during evaluation, we found that MobileNetV2 did not meet our computational efficiency requirements as effectively as EfficientNet B0. EfficientNet B0's optimized architecture, with approximately 5.3 million parameters and 0.39 GFLOPS, aligned better with the project's real-time inference requirement, providing a high-performance solution capable of delivering accurate predictions within the specified timeframe.

The original classification layer of EfficientNet B0 was fine-tuned to perform binary classification by adapting it to output a single logit, representing the probability of alignment. EfficientNet B0 expects images that have undergone specific preprocessing transformations, including resizing to 256 pixels, central cropping to 224 pixels, and normalization with mean values [0.485, 0.456, 0.406] and standard deviations [0.229, 0.224, 0.225].

Results

Our model achieved strong performance on the validation dataset, with optimal results observed at epoch 10 during training. Although the model was trained for 30 epochs, heuristic analysis indicated that performance peaked at epoch 10, after which the model began to overfit. Below are the key metrics for epoch 10:

- **Accuracy:** 0.9340
- **Precision:** 0.9437
- **Recall:** 0.9582
- **F1 Score:** 0.9509
- **F-beta Score** (with $\beta = 0.5$): 0.9466

The **F-beta score** is used to balance precision and recall, prioritizing precision with $\beta = 0.5$. It is calculated as follows:

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

where:

- **Precision:** The proportion of true positives among all predicted positives.
- **Recall:** The proportion of true positives among all actual positives.

The confusion matrix below summarizes the classification results at epoch 10:

	Predicted Aligned	**Predicted Not Aligned**
Actual Aligned	201	26
Actual Not Aligned	19	436

Table 1: Confusion Matrix for Epoch 10

In addition to classification performance, inference time was a critical consideration due to the project's real-time requirement. The following table shows the key statistics for inference time per image:

Inference Time Statistic	**Time (seconds)**
Mean Time	0.0298
25th Percentile	0.0111
Median (50th Percentile)	0.0117
75th Percentile	0.0128

Table 2: Inference Time Statistics Per Image

These results indicate the model's capability to accurately distinguish between aligned and not-aligned samples while achieving real-time inference performance, meeting the project's requirement of under 0.5 seconds per image on commercial hardware.

Challenges

Throughout the project, we encountered several challenges related to dataset variability and performance requirements:

- **Lighting and Perspective Variations:** The images in the dataset were captured under varying lighting conditions and from different camera perspectives, which can often complicate alignment detection. However, the use of a visual transformer (DINO v2) proved highly effective in handling these inconsistencies. Its self-supervised learning capabilities allowed it to capture robust visual features, making it more resilient to lighting and perspective shifts.
- **Inference Time Optimization:** To meet the real-time inference requirement on commercial hardware, we carefully selected EfficientNet B0 for its computational efficiency. This model allowed us to maintain high performance while staying within the required inference time of less than 0.5 seconds per image.

References

[1] Oquab, Maxime et al., *Dinov2: Learning robust visual features without supervision*, arXiv preprint arXiv:2304.07193, 2023.

[2] Tan, Mingxing and Le, Quoc, *EfficientNet: Rethinking model scaling for convolutional neural networks*, International Conference on Machine Learning, pages 6105–6114, 2019, PMLR.