ConveyorVision Bag Counter

Jagennath Hari

***Abstract*—ConveyorVision is an innovative real-time system designed to automate the counting and tracking of cement bags on conveyor belts. Utilizing cutting-edge deep learning techniques like YOLOv8 for object detection and Byte tracker for precise tracking, ConveyorVision accurately monitors cement bags as they traverse the conveyor belt. Its seamless integration, reliable counting at the referee line, and robust performance in complex environments make it a valuable tool for optimizing industrial processes and enhancing productivity.**

***Index Terms*—Computer vision, Deep learning, Object detection, Image processing, Precision tracking, Industrial automation**

I. Introduction

In the realm of modern industries, automated systems have become indispensable for improving efficiency, accuracy, and productivity. Conveyor belts, serving as the backbone of many manufacturing and logistics processes, require effective monitoring and control to ensure seamless operations. Specifically, in the cement industry, the precise counting and tracking of cement bags as they traverse conveyor belts play a crucial role in optimizing production and logistics.

Traditionally, proximity switches shown in Figure 1 have been employed for bag counting, but this approach has its limitations. It lacks the ability to distinguish individual bags and may not be reliable in dynamic scenarios with overlapping bags. Moreover, manual counting, though more accurate, is labor-intensive, time-consuming, and susceptible to human errors. Advancements in computer vision and deep learning have paved the way for sophisticated automation solutions that can significantly enhance industrial processes.

"ConveyVision," a state-of-the-art real-time cement bag counting and tracking system designed to address the limitations of manual methods. Leveraging cutting-edge technologies, including YOLOv8 for object detection an example in Figure 2 and Byte tracker for accurate tracking, ConveyorVision offers a comprehensive and efficient solution for industrial monitoring.

The primary objective of ConveyorVision is to automate the counting process and provide precise tracking information for each cement bag moving on the conveyor belt. The integration of YOLOv8 allows the system to detect cement bags with high accuracy, even in challenging and dynamic environments. By identifying individual bags, ConveyorVision can precisely track their positions and movements using the Byte tracker algorithm, ensuring continuous and reliable monitoring.

Furthermore, ConveyorVision is designed for seamless integration into existing conveyor belt setups, making it an accessible and practical solution for industries seeking process optimization. The system's real-time capabilities enable instant decision-making and facilitate proactive adjustments to improve production efficiency.

Fig. 1: Example of a traditional system.

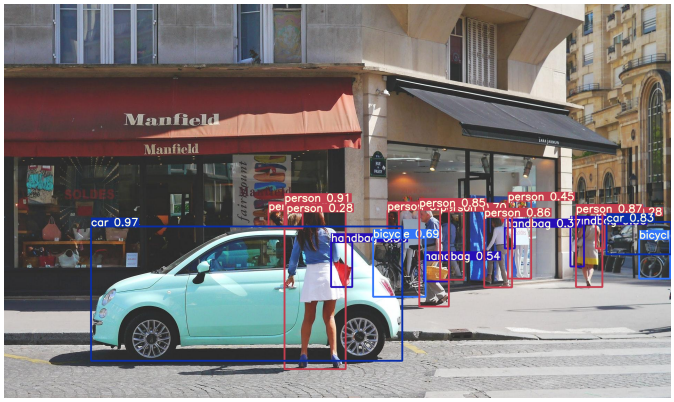

Fig. 2: Object detection using YOLOv8.

In this report, we present the design and implementation of ConveyorVision, providing insights into its key components and how they collaborate to achieve accurate counting and tracking results. We also report on extensive experiments conducted to evaluate the system's robustness in diverse industrial scenarios.

The rest of the report is organized as follows: Section II

offers a comprehensive understanding of the neural network. Section II also details the methodology employed in building ConveyorVision, including the technical specifications and algorithms used. Section III presents the experimental setup and discusses the results obtained from various scenarios. Finally, Section IV concludes the report, showcasing the results of ConveyorVision and while indicating a few limitations.

With ConveyorVision, we aim to contribute to the growing field of smart manufacturing and industrial automation, providing industries with a valuable tool to optimize processes, enhance productivity, and usher in a new era of efficiency and accuracy in cement bag counting and tracking.

II. Methodology

In this section, the neural network in Section II-A for object detection is examined along with the technique used for tracking in Section II-B. Finally, the entire algorithm is enumerated in Section II-C.

A. Neural Network

You Only Look Once (YOLO) is a groundbreaking object detection algorithm that revolutionized the field of computer vision. YOLO changed the conventional approach to object detection by introducing a single-stage, real-time detection method.

Traditional object detection methods involved two stages: region proposal and object classification. These methods were often computationally expensive and time-consuming, limiting their applicability in real-time scenarios. YOLO, on the other hand, eliminated the need for region proposal and directly predicted bounding boxes and class probabilities in a single forward pass through the neural network.

The core idea behind YOLO shown in Figure 3 is to divide the input image into a grid and predict bounding boxes for objects within each grid cell. Each bounding box contains information about the object's coordinates, width, height, and class probabilities. The predictions are made at multiple scales to detect objects of different sizes in the image.

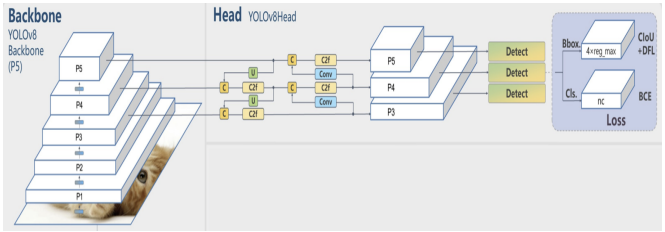

Fig. 3: YOLOv8 network architecture.

One of the key advantages of YOLO is its impressive speed. By simultaneously predicting bounding boxes and class probabilities, YOLO achieves real-time object detection on standard hardware, making it highly suitable for applications requiring fast and accurate detection, such as autonomous vehicles, surveillance, and robotics.

YOLO's real-time performance, coupled with its ability to capture small objects and handle overlapping instances, has made it a popular choice for various computer vision tasks. Several improved versions of YOLO, such as YOLOv2 and YOLOv3, have been introduced to further enhance accuracy and speed.

Despite its strengths, YOLO does have some limitations, particularly in detecting small objects and accurately localizing objects near the image boundaries. Nevertheless, YOLO's groundbreaking design and its impact on real-time object detection have inspired numerous research advancements in the computer vision community.

B. Tracking

Multi-object tracking (MOT) is a fundamental task in computer vision that involves detecting and tracking multiple objects in video sequences. However, traditional MOT methods often suffer from issues like true object misses and fragmented trajectories, primarily due to the handling of low-score defections. These defections, such as occluded objects, are typically discarded, leading to a loss of valuable information and performance degradation.

To tackle these challenges and improve the effectiveness of MOT, a team of researchers proposes a novel and robust association method. Unlike conventional approaches that only consider high-score detection boxes, this new method associates every detection box during the tracking process. By doing so, the team aims to retain valuable information from low-score detection's, which may represent important but occluded or partially visible objects shown in Figure 4.

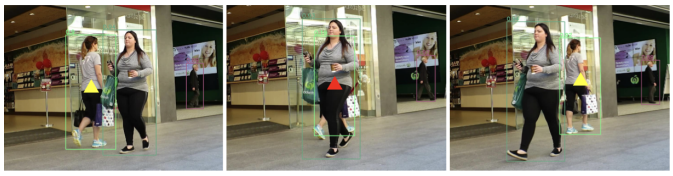

Fig. 4: Tracking during occlusion.

The key innovation of the proposed method lies in leveraging the similarities between low-score detection's and existing tracklets. Tracklets are short track segments that correspond to previously tracked objects. By analyzing the similarities between these tracklets and the low-score detection's, the method can recover true objects and distinguish them from background detection's effectively.

C. Algorithm

A video is typically a sequence of images, the entire algorithm runs on a frame to frame(F2F) basis where the network and tracker are deployed across a sequence of frames. The F2F approach is commonly used in the computer vision and robotics community to establish sequential information using time-series.

Let us consider frame 1 as F_0, frame 2 as F_1, frame 3 as F_2 and, so forth. The object detection for a frame $f(F_x)$ outputs a descriptor vector b_x which contains the pixel coordinates of objects in F_x. The center point of b_x is used

to perform tracking in F_{x+1} using a loss function $l(b_{x+1}, b_x)$. The loss value $L2$ which normally is the Euclidean distance in pixel space is filtered out using a threshold, typically Lowe's distance test. Given the loss $L2$ is less than Lowe's value t, the match is accepted between F_{x+1} and F_x. The descriptor for the correspondence in F_{x+1} is partially replaced using F_x, while keeping the pixel coordinates of the descriptor the same. This continues until F_∞, where the loss/cost function is used to maintain tracking. In cases during occlusion, where the b_x in null, b_{x-n} where n satisfies Lowe's test is used to maintain target tracking. The aforementioned algorithm is better shown in Algorithm 1.

Algorithm 1 Object detection and tracking (F_x, F_{x+1})

Output: Location of tracked target

1: **for** F_x in video **do**
2: $F_{x+1} \leftarrow$ `Object detection`(F_x)
3: $F_{x+1} \leftarrow l(b_{x+1}, b_x)$
4: **if** `Similarity`$(b_{x+1}, b_x) < L2$ **then**
5: Replace b_x with b_{x+1}
6: **else**
7: Search for correspondence between b_{x-n} and b_{x+1} where n satisfies $L2$
8: **end if**
9: **end for**

D. Reference line for improved robustness

Algorithm 1 shows how tracking is performed for objects in a video, the tracked object can be assigned IDs to ensure the total objects in a video are determined. To improve the accuracy two reference points or a line can be incorporated into the image to evaluate whether the tracked object has crossed the reference line, giving rise to improved robustness. An arbitrary reference line is added in an image, when b_{x+1} crossed the reference line's pixel location the count is incremented, this continues until F_∞. The reference line being arbitrary pixel values can be used to track and adjust the count values when an object also moves in the opposite direction.

III. Implementation

The network initially needs to be trained on images on the bag, in order for it to learn the unique features which classify it as a bag as shown in Section III-A. The models department is explained in Section III-B.

A. Training

Most neural networks for modern computer vision application are trained using a GPU, in this setup a NVIDIA RTX 4050 GPU was used to train the network.

The network was trained to close to 250 epochs, to allow the Stochastic gradient descent to converge to the box loss. The results after training the network are shown in Figure 5.

Figure 6 shows the performance on the validation dataset, showing the network has converged optimally for real-time deployment.

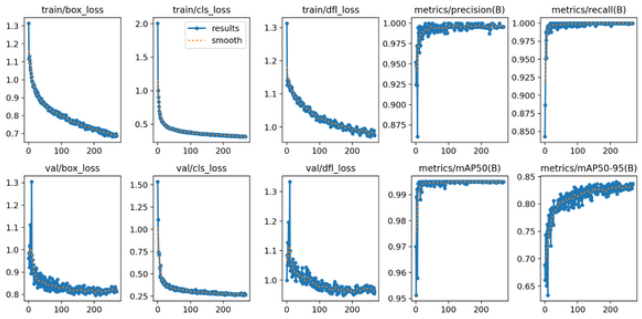

Fig. 5: Training curves.

Fig. 6: Validation of network.

B. Deployment

The deployment stage uses the fully trained network in Section III using a F2F estimation, along with tracking in Section II-B. A reference line as indicated in Algorithm 1 is added to improve the robustness for dynamic environments, when several occlusion may occur when using the F2F approach. Figure 7 shows an instance of the entire Algorithm 1, "out" represents the number of bag crossed the reference line and "in" was added just in case the the conveyor ever travels in the opposing direction. The numerical value on each bag shown in Figure 7 depicts the confidence of the network when classifying the object, in our case "bag".

C. ConveyorVision Pipeline

Figure 8 depicts the pipeline, from Algorithm 1, F_0 and F_1 are Image 0 and Image 1, respectively. The Neural network computes the descriptor b for each image, when these descriptor fed as input in the tracking, matches their correspondences giving rise to the tracked IDs of the target object. When done

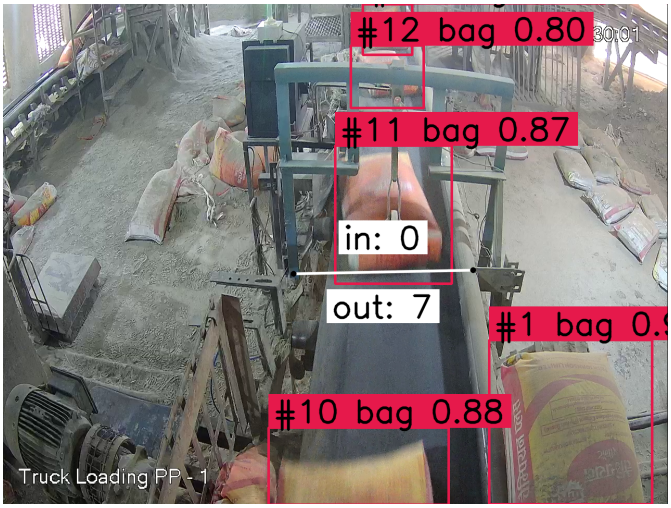

Fig. 7: ConveyorVision deployment.

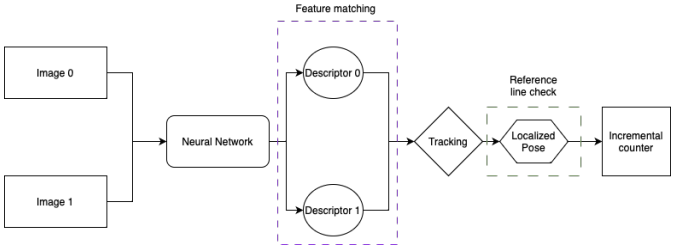

Fig. 8: Pipeline.

sequential across multiple frames in a video, the algorithm maintains the tracked targets along with IDs until the target is no longer within the frame. Time-series and expanding the tracking across a local region across multiple frames ensures that tracking stays intact during occlusion in dynamic environments.

IV. Results

TABLE I: Performance of the network on the videos.

Dataset	Actual number of bags	ConveyorVision count	Average time in ms/frame
1	413	413	4.3 ms
2	287	287	5.6 ms

Table I shows the results when employed on the two datasets provided. The accuracy of the network was 100% at $4.8ms$ per frame, indicating the model can be deployed in real-time in dynamic environments where occlusion, or when bags are close to one another, which typically causes traditional systems to fail. The only limitation of this system is during camera failure or where portions of the video stream has been lost due to unknown reasons, these anomalies can give rise to a few false positives or miss-counts of the cement bags.