MISSING BOLT DETECTION ON A CONVEYOR BELT

Hemalatha S[1], Devadharshini C[2], KavyaSruthi S S[2], HafidWaqar Ahmed S[2], Sudharma K V[2]
[1]HOD, Dept. of AIML
Sri Shakthi Institute of Engineering and Technology, Coimbatore
[2]Dept. of AIML
Sri Shakthi Institute of Engineering and Technology, Coimbatore

Abstract: *As an essential component of conveyor systems, ensuring the integrity of conveyor belts is vital for safe and efficient operations. Missing bolts can cause equipment failure, operational delays, and safety hazards. Traditional manual inspections are labor-intensive, inefficient, and prone to errors, especially in harsh environments like heavy dust or poor lighting. This paper introduces a missing bolt detection system using YOLOv8, a cutting-edge deep learning algorithm. YOLOv8's advanced features enable the accurate detection of missing bolts on conveyor components. The model is trained with annotated images captured under challenging conditions, including dust, low light, and partial obstructions. The system analyzes real-time video or images from industrial cameras to locate missing bolts. Detected issues are marked immediately, allowing timely corrective actions. Field tests show high accuracy and reliability, even in complex conditions. This solution significantly reduces risks associated with missing bolts and enhances maintenance efficiency. It ensures conveyor system safety and operational reliability in demanding industrial environments.*

Keywords: Missing bolts,YOLOv8, Conveyor belt, real-time detection.

I. INTRODUCTION

Conveyor belts play a critical role in industrial processes, ensuring efficient material transportation and operational continuity. Missing bolts in conveyor belt structures can lead to equipment failures, unplanned downtime, and safety hazards, severely impacting productivity. According to industry statistics, structural failures caused by undetected missing bolts are a significant concern in industrial operations. These issues highlight the need for an accurate, reliable, and real-time bolt detection system.

Traditional approaches to detecting missing bolts, such as manual inspection and metal detectors, have notable limitations. Manual inspection relies heavily on human observation, which is prone to fatigue and errors, especially over long durations and in harsh environments with low light or heavy dust. While metal detectors can detect certain anomalies, they struggle with non-metallic components and cannot localize missing bolts effectively. Similarly, imaging-based methods using conventional object detection techniques provide limited accuracy in complex industrial settings.

The advent of deep learning and computer vision technologies has paved the way for innovative solutions in object detection. YOLOv8, as a cutting-edge deep learning framework, offers outstanding accuracy and real-time performance, making it ideal for detecting missing bolts on conveyor belts in industrial environments. By leveraging YOLOv8’s advanced feature extraction capabilities, this paper proposes a detection method tailored for the industrial domain.

The key contributions of this research are as follows:
1. Proposed Detection System: A bolt detection system using YOLOv8 is developed, enabling real-time detection of missing bolts in conveyor belt structures with high accuracy.

2. System Design: The detection system incorporates robust image preprocessing and a streamlined model architecture to ensure reliable operation in challenging environments, such as low light, dust, and occlusions.

3. Performance Optimization: The system is optimized for portable, low-cost hardware setups by reducing computational overhead while maintaining detection precision and speed.

II. LITERATURE REVIEW

Accurate detection of structural anomalies in conveyor systems is crucial for ensuring safety and operational efficiency in industrial environments. Various approaches have been explored in the past for detecting structural and material defects, including bolt detection and other foreign object recognition tasks.

Traditional detection methods, such as manual inspections, rely on workers visually identifying anomalies along conveyor belts. While this approach is simple, it suffers from significant limitations, including high labor costs, inefficiency, and vulnerability to human fatigue and environmental factors such as dust and poor lighting. These drawbacks lead to low detection accuracy, particularly in industrial environments where conveyor belts span long distances.

Metal detection techniques have been used to identify metallic objects such as bolts or scrap iron along conveyor belts. These systems employ strong magnets or metal sensors to attract and detect metallic items. However, they are ineffective in detecting non-metallic components and fail to localize missing bolts accurately. Additionally, buried or partially concealed bolts are often undetectable, limiting the method's utility.

Ray detection methods, including γ-rays and X-rays, distinguish materials based on their energy absorption coefficients. While effective for identifying foreign objects, these techniques are costly, require regular maintenance, and expose operators to potentially harmful radiation. These factors restrict their widespread industrial adoption.

The advent of machine vision technologies has transformed defect detection by providing high accuracy and real-time performance. Early applications included conventional image processing techniques for material classification and defect identification. For instance, Zhao et al. [1] developed a coal gangue recognition system using image processing, while Le et al. [2] proposed a gray-scale compression method for identifying material anomalies. These systems demonstrated potential in structured environments but struggled with complex industrial conditions involving dust, poor lighting, and occlusions.

Recent developments in deep learning have significantly enhanced object detection capabilities. Convolutional Neural Networks (CNNs) have been widely adopted for their ability to extract and process complex features from images. Zhang et al. [3] introduced a CNN with attention modules to segment foreign objects from noisy backgrounds. Although the model achieved high accuracy, its dependence on GPU resources increased system cost and complexity. Similarly, Wang et al. [4] applied an SSDbased video method for detecting surface-level anomalies, but its performance was hindered by high computational requirements.

To address these limitations, researchers have proposed lightweight and efficient detection models. Xiao et al. [5] designed a model combining RDU-Net with CNN residual structures to improve accuracy while reducing computational overhead. Gaurav Saran et al. [6] utilized multi-modal imaging techniques to enhance detection robustness but faced challenges in real-time performance due to high latency.

Building on these advancements, YOLO (You Only Look Once) frameworks have emerged as state-of-the-art solutions for real-time object detection. YOLOv8, the latest in the series, offers exceptional detection speed and accuracy while maintaining a compact model size. Its feature extraction and multi-scale prediction capabilities make it highly suitable for detecting small and intricate objects, such as missing bolts on conveyor belts.

Despite the extensive progress, existing approaches for conveyor belt inspection often face challenges in balancing accuracy, real-time performance, and cost-effectiveness. This research leverages YOLOv8 to address these challenges, providing a robust and efficient system for missing bolt detection under complex industrial conditions.

III. EXISTING SYSTEM

The system for detecting missing bolts on conveyor belts utilizes synthetic data, domain adaptation, and advanced machine vision techniques. Synthetic data is generated using a simulator, annotated for segmentation, and used to train the YOLOv8s-seg model. A CycleGAN network addresses the domain gap by performing style transfer between synthetic and real-world data. The methodology includes four stages: data collection and preprocessing, training YOLOv8s-seg on synthetic data, domain adaptation via CycleGAN, and fine-tuning the model with style-transferred data. Real-world data is collected using a 3DPM test rig, capturing images of conveyor belts with anomalies like reshaped cables and rocks. These images are processed, annotated, and formatted for YOLOv8 segmentation. The YOLOv8s-seg model, pre-trained on the COCO dataset, is fine-tuned using 3759 training images and evaluated on validation and test sets. Training is conducted over 200 epochs with SGD optimization and data augmentation techniques to enhance robustness. The final domain-adapted model effectively detects anomalies on real-world conveyor belts.

DISADVANTAGES

Performance Gap for Boulder Anomalies: There is a significant domain gap between synthetic and real-world data for the boulder anomaly, resulting in much lower performance metrics (mAP scores).

Fixed Density in Synthetic Data Simulation: The simulator uses a fixed density for anomalies, which affects the realism of anomaly distribution and the frequency of their appearance, potentially leading to less effective training data.

Limited Diversity in Synthetic Data: The simulator lacks diversity in particulate matter and anomaly types, limiting the model's ability to generalize to broader scenarios.

Annotation Challenges for Occluded Objects:

Annotations for partially occluded objects or objects with disconnected parts could be improved to better represent the anomalies.

Limited Hardware Capabilities: The CycleGAN **training** resolution was constrained by the VRAM (8GB) of the AMD Radeon™ RX 6600, restricting the quality of style transfer.

Insufficient Real-World Data: The limited amount of real-**world** data affected the domain adaptation process and required lowering the learning rate to avoid mode collapse.

IV. PROPOSED SYSTEM

The proposed system for detecting missing bolts on industrial conveyor belts ensures accuracy, scalability, and adaptability across diverse environments. Synthetic data generation simulates various bolt types, backgrounds, and lighting conditions, reducing dependency on real-world data. Consistent annotation and preprocessing prepare the data for YOLOv8's instance segmentation, enhancing robustness. CycleGAN-driven domain adaptation bridges the synthetic-to-real-world data gap, enabling the YOLOv8 model to generalize effectively. The fine-tuned model is deployed for real-time detection, utilizing GPU optimization for high-speed inference. API integration enables seamless interaction with monitoring systems, providing continuous anomaly detection. This automation reduces maintenance costs, prevents downtime, and improves operational safety. Future enhancements could include expanded synthetic datasets, environmental simulations, and explainability tools like Grad-CAM to improve trust and transparency. The system's real-time capabilities make it suitable for various industrial applications.

ADVANTAGES

- **Scalability**: The use of synthetic data reduces dependence on labor-intensive real-world data collection, enabling the model to be trained for various scenarios.
- **Real-Time Capability**: High inference speeds allow for continuous monitoring of conveyor belts without interrupting industrial operations.
- **Adaptability**: Domain adaptation via CycleGAN ensures that the system performs effectively in diverse environments, including varying lighting and surface conditions.
- **Precision and Robustness**: The YOLOv8 algorithm's advanced architecture ensures accurate detection of missing bolts, reducing false positives and negatives.

V. MODEL DESIGN

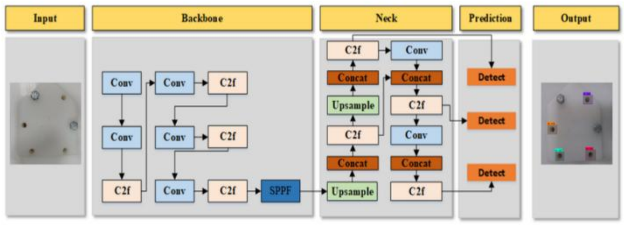

Figure: Model Design

INPUT:

- The process begins with an image input (e.g., an image of a surface containing bolts).
- This image serves as the raw data that the model will analyze to detect the presence or absence of bolts.
- Example Input: An image containing bolts and possibly empty bolt positions.

BACKBONE:

- This is the feature extraction part of the model, responsible for extracting essential visual features from the input image.
- Conv: Convolutional layers that apply filters to detect patterns, textures, and edges in the image.
- C2f: A block (possibly "CSP2f" from YOLO-like architectures) used for enhanced feature extraction while optimizing computation. It splits the feature map into two parts: one undergoes transformations, and the other skips them, improving gradient flow and efficiency.
- SPPF (Spatial Pyramid Pooling Fast): Aggregates features at multiple scales, enabling the model to better capture objects of varying sizes (such as bolts in different positions).

NECK:

- This stage is responsible for feature aggregation and refining, enabling the model to integrate feature maps at different resolutions and scales.
- Upsample: Upsampling increases the spatial resolution of feature maps to match dimensions for concatenation.
- Concat: Concatenates feature maps from different levels of the architecture to preserve multi-scale information.
- C2f: Additional feature extraction blocks that further refine the aggregated features.
- Purpose: This stage combines high-level and low-level features to localize and identify objects more effectively.

PREDICTION:

- This stage generates predictions about the objects present in the image, including their locations (bounding boxes) and categories (e.g., "bolt" or "missing bolt").
- Detect: The detection head applies learned parameters to predict object classes and their bounding boxes. It uses the processed feature maps from the neck to identify objects.
- Outputs: Bounding boxes and class labels for the detected objects.

OUTPUT:

- The final output displays the image with detected objects labeled, along with bounding boxes.
- Bolts are correctly identified and labeled.
- Missing bolts (empty positions) are identified and flagged as "missing."

HARDWARE REQUIREMENTS

- CPU type : Intel Pentium 4
- Clock speed : 3.0 GHz
- Ram size : 4 GB
- Hard disk capacity : 100 GB
- Monitor type : 15 inch colour monitor
- Keyboard type : internet keyboard

SOFTWARE REQUIREMENTS

- Operating System : Windows OS
- Front End : PYTHON

VI. RESULT

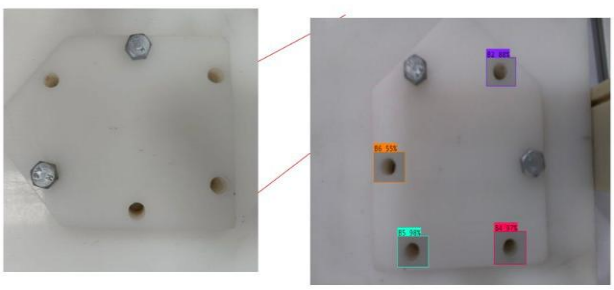

Figure: Detection of Missing Bolts

The provided image illustrates the process and results of a missing bolt detection system. The left side of the image represents the input scene, while the right side demonstrates the output of the detection system, where bolts and missing positions are identified and labeled with bounding boxes and confidence scores. Here's a detailed explanation of the process:

Input:

- The system takes an image of a mechanical surface with bolt positions as input.
- In this particular example, some bolts are properly fixed in their positions, while others are missing.

Model Processing:

- The image is passed through a trained object detection model, which analyzes it in real-time to identify the presence or absence of bolts at specific locations.
- The model processes the input image using a feature extraction and prediction pipeline (as explained earlier in the architecture).
- It identifies:
  - **Bolts**: Fully secured and visible bolts are detected.
  - **Empty Positions**: Positions where bolts are missing are also identified.

Detection and Labeling:

- The processed output (right side of the image) shows the detection results:
- **Bounding Boxes**: The system draws colored boxes around identified bolts and empty positions to indicate their locations.
- **Confidence Scores**: Each bounding box is annotated with a percentage, representing the model's confidence in the detection.
- For example: **B5 (98%)**: Indicates a detected empty bolt position with a 98% confidence score and **B2 (88%)**: Indicates a secured bolt detected with 88% confidence.
- The labeled results provide a clear distinction between present bolts and missing bolts.

Decision Making:

- Based on the detection results, the system categorizes positions as: **Secured**: Bolts properly installed, as shown in positions like B2 and **Missing/Defective**: Empty positions where bolts are absent, such as B6 (confidence 55%).
- The results can trigger appropriate actions: **Alert**: If missing bolts are detected, the system generates an alert and **Halt Process**: The production or inspection process can halt to address the defect.

Output Visualization:

- The final output provides a comprehensive overview of the bolt inspection: Users can easily identify which positions are missing bolts or need attention and The high-confidence labels ensure reliability in quality control tasks.

VII. CONCLUSION

The project on missing bolt detection on industrial conveyor belts using YOLOv8 and API integration demonstrates a robust, scalable, and cost-effective approach to anomaly detection. By leveraging synthetic data generation, the system effectively addresses the challenge of limited real-world anomalous datasets. The use of CycleGAN-driven domain adaptation ensures that the YOLOv8 model generalizes well from synthetic to real-world conditions, improving accuracy in diverse environments. The integration of an API key allows seamless real-time communication between the detection model and conveyor belt monitoring systems, enabling continuous and automated monitoring without manual intervention. The high inference speeds achieved make the system suitable for real-time applications, reducing maintenance costs, preventing operational downtime, and ensuring safety. This project exemplifies a cutting-edge application of deep learning and domain adaptation techniques in industrial settings.

REFERENCES

[1]. Algoryx. Agx dynamics. https://www.algoryx.se/agx-dynamics/, 2024. [Accessed 2024-05-21].

[2]. Shah Deval. Mean average precision (map) explained: Everything you need to know. https://www.v7labs.com/blog/mean-average-precision, 2022. [Accessed 2024-05-19].

[3]. YaroslavGanin, EvgeniyaUstinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, Franc¸oisLaviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks, 2016.

[4]. Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. A neural algorithm of artistic style, 2015.

[5]. Ian Goodfellow, YoshuaBengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[6]. Kaiming He, Xiangyu Zhang, ShaoqingRen, and Jian Sun. *Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition*, page 346–361. Springer International Publishing, 2014.

[7]. Kaiming He, Xiangyu Zhang, ShaoqingRen, and Jian Sun. Deep residual learning for image recognition, 2015.

[8]. Hermary. Principles of laser triangulation. https://hermary.com/learning/ principles-of-laser-triangulation/. [Accessed 2024-05-20].

[9]. IBM. What are support vector machines (svms)? https://www.ibm.com/topics/ support-vector-machine. [Accessed 2024-05-20].

[10]. IBM. What is random forest? https://www.ibm.com/topics/random-forest. [Accessed 2024-05-20].

[11]. learnopencv.Sgd.https://learnopencv.com/ultralytics-yolov8/, 2024. [Accessed 2024-05-21].

[12]. Zeming Li, Chao Peng, Gang Yu, Xiangyu Zhang, Yangdong Deng, and Jian Sun. Light-head r-cnn: In defense of two-stage object detector, 2017.

[13]. Tsung-Yi Lin, Michael Maire, Serge Belongie, LubomirBourdev, Ross Girshick, James Hays, PietroPerona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollar.´ Microsoft coco: Common objects in context, 2015.

[14]. Vladimir Nasteski. An overview of the supervised machine learning methods. *HORIZONS.B*, 4:51–62, 12 2017.

Author Profile

Mrs. S. Hemalatha, has over 10 years of experience in academics and machine learning research. **Devadharshini C, KavyaSruthi S S, HafidWaqar Ahmed S,** and **Sudharma K V**are final-year AIML students at Sri Shakthi Institute of Engineering and Technology, specializing in computervision and deep learning applications.