KLE Society's

KLE Technological University

An Industry Project Report

On

Machine Learning-Based Real-Time Conveyor Inspection and Performance Improvement for Steel Industries

Submitted in partial fulfillment of the requirement for the degree of

Bachelor of Engineering in

Computer Science and Engineering

Submitted By

Shreyas Joshi
01fe20bcs145

Under the guidance
of

Mrs. Pratibha R Malagatti

SCHOOL OF COMPUTER SCIENCE &ENGINEERING,

HUBBLLI–580 031 (India).

Academic year 2023-24

B. V. Bhoomaraddi College Campus, Vidyanagar, Hubballi - 580031.
Karnataka (India)

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that Industry Project entitled **"Machine Learning-Based Real-Time Conveyor Inspection and Performance Improvement for Steel Industries"** is a bonafide work carried out by the student **Mr. Shreyas Joshi** bearing USN **01fe20bcs145** in partial fulfillment of the completion of 8th semester B. E. course during the year 2023 – 24 at **DocketRun Tech Private Limited**. The Industry Project report has been approved as it satisfies the academic requirement with respect to the project work prescribed for the above said course.

Name of the Guide	**Head of SoCSE**
Mrs. Pratibha R Malagatti	**Dr. Vijayalakshmi M.**

Name of the examiners	**Signature with date**
1 ------------------------- -----------------	1 ------------------------- -----------------
2 ------------------------- ----------------	2 ------------------------- ----------------

DOCKETRUN
TECH PRIVATE LIMITED

DOCKETRUN TECH PRIVATE LIMITED

1st Floor, R.H. Kulkarni Building, KLE Tech University, Vidyanagar Hubballi Karnataka – 580031
CIN: U72900KA2019PTC128107 GST: 29AAHCD4356L1Z6 PAN: AAHCD4356L
Mob: +91-9449034387 Email: info@docketrun.com Website: www.docketrun.com

INTERNSHIP COMPLETION CERTIFICATE

We are delighted to certify that **Mr. Shreyas Joshi,** a student of Computer Science Engineering, **BVB College, KLE Tech University, Vidyanagar Hubballi**, has successfully completed the internship program (08/01/2024 – 31/05/2023) at **DocketRun Tech Pvt Ltd.**

During the internship, he played a pivotal role in multiple projects, focusing on AI development & completed numerous assignments, showcasing adaptability and excellence. Notably, he was a key contributor to the **"Machine Learning-Based Real-Time Conveyor Inspection and Performance Improvement for Steel Industries"** project.

We would like to express our sincere gratitude for your exemplary performance and wish you continued success in all your future endeavors.

Warm Regard's

DOCKETRUN TECH PRIVATE LIMITED HUBBALLI

Ajay S Kabadi

Ajay S Kabadi
Founder, C.E.O
DocketRun Tech Pvt Ltd.

DECLARATION

I hereby declare that the Industry Project Report entitled **"Machine Learning Based Real-Time Conveyor Inspection and Performance Improvement for Steel Industries"** is an authentic record of my own work as requirements of Industry, during the period from 08-01-2024 to 31-05-2024 for the award of the degree of B.E. Under the guidance of Mrs. Pratibha R Malagatti.

(Signature of student)

Shreyas Joshi
01fe20bcs145

DATE:

Acknowledgement

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of a number of individuals whose professional guidance and encouragement helped me in the successful completion of this report work.

I also take this opportunity to thank Dr. Vijaylakshmi M, Professor and Head, School of Computer Science and Engineering for having provided us academic environment which nurtured our practical skills contributing to the success of our project.

I sincerely thank our guide Mrs. Pratibha R Malagatti, School of Computer Science and Engineering for her guidance and wholehearted co-operation during the course of completion.

I sincerely thank Mr. Huchchareddi, Docketrun for his support, inspiration and wholehearted co-operation during the course of completion.

My gratitude will not be complete without thanking our beloved parents, our seniors and our friends who have been a constant source of aspirations.

Shreyas Joshi

ABSTRACT

Video analytics, powered by advancements in artificial intelligence and machine learning, is transforming the landscape of video data analysis. This technology enables real-time detection, tracking, and analysis of objects, faces, and activities in video streams. Key tools such as OpenCV and YOLO (You Only Look Once) are instrumental in enabling real-time processing of video data, automating tasks, and enhancing decision-making processes. The manufacturing sector in India faces a critical challenge in ensuring the safety and well-being of its workforce, as evidenced by alarming statistics indicating a significant number of fatalities and injuries reported annually. This underscores the urgent need for enhanced safety and security measures in factories to mitigate risks and improve workplace safety. Implementing video analytics for real-time monitoring offers a comprehensive approach to addressing these challenges in the manufacturing sector. By harnessing the power of machine learning, artificial intelligence, image processing, and video data processing, technologies like OpenCV and YOLO can provide instantaneous analysis directly on the factory floor. This application goes beyond traditional safety measures, offering a range of benefits including enhanced safety and security, downtime reduction, risk minimization, cost savings, remote monitoring. Here we provide end-to-end machine learning system for real-time monitoring and inspection of CCTV inputs for conveyor machines in the manufacturing sector. The system's objective is to enhance operational performance and ensure workplace safety by leveraging state-of-the-art video analytics technology. We employ wide range of components such as video processing, object detection using YOLOv8. The choice of technologies and libraries, such as Python for programming, OpenCV for video processing, and YOLOv8 for object detection, are made based on their suitability for the project requirements. The system's user interface, including any GUI elements for monitoring and configuration, are designed using python's library Tkinter. Additionally, considerations are made for performance optimization, security, scalability, and reliability.

Keywords : *YOLO, OpenCV, ROI, Object detection, Video analytics, CCTV, SSD*

CONTENTS

LIST OF FIGURES

Chapter 1

INTRODUCTION

Video analytics is a rapidly evolving field that leverages artificial intelligence and machine learning to extract valuable insights from video data. By analyzing video streams in real time, video analytics systems can detect, track, and analyze objects, faces, and activities, enabling a wide range of applications in security, surveillance, retail, and more. Tools and technologies such as OpenCV and YOLO (You Only Look Once) play a crucial role in enabling real-time processing of video data, making it possible to automate tasks and enhance decision-making processes. Among various machine learning algorithms available, the Single Shot Detector (SSD) method, based on convolutional neural networks (CNNs), stands out for its real-time object detection capabilities. The primary objective of this approach is to enhance the accuracy of the SSD method for object detection tasks. Known for its speed and high accuracy, even with low-resolution images, the SSD method eliminates the need for a separate region proposal network, thereby improving efficiency. Additionally, the SSD method can detect multiple objects simultaneously within an image, making it ideal for applications such as video surveillance, autonomous vehicles, and industrial automation. One such example of SSD is YOLO.Applications of YOLO object detection model include multi-object tracking, event detection, scene understanding, crowd analysis, and searching video databases. It highlights the chal- lenge posed by the overwhelming volume of raw video data generated by surveillance cameras, driving the need for intelligent video analytics to extract useful information automatically.

This project is an end-to-end machine learning solution designed to enhance industrial safety and operational efficiency by reducing downtime, automating monitoring processes, and improving system performance. The innovative approach focuses on real-time monitoring and detection of safety violations in live video feeds, specifically targeting issues such as missing cleats and non-linear walls. In steel industries, both missing or unaligned cleats and non-linear walls pose significant challenges to the smooth operation of machinery. These issues can lead to downtime and decreased efficiency, which our project aims to mitigate using advanced technologies. We utilize YOLO (You Only Look Once) for object detection and the OpenCV Python library for image processing to develop an intelligent monitoring system that detects and records safety violations in real-time. The developed algorithm automates the monitoring process, making it easier to identify violations and thereby reducing downtime. It works by processing input from video or live feeds, extracting individual frames, and analyzing them. The YOLO object detection model is applied to each frame to predict the coordinates of missing or unaligned cleats and to identify any safety violations. This method ensures high accuracy and efficiency in real-time detection. For detecting non-linear walls, we employ multiple image filters applied in sequence using OpenCV. This approach effectively identifies deviations from linearity, which are crucial for maintaining smooth operations in steel industries.

Our system not only provides real-time monitoring of the live feed but also records and stores detected violations. When a violation, such as a missing cleat or a non-linear wall, is identified, the system captures a snapshot of the relevant frame. This data is then saved locally and in a database, allowing for comprehensive analysis and record-keeping. By automating the detection and recording process, our solution enhances operational efficiency and safety in industrial environments. The stored data facilitates ongoing analysis and helps in maintaining a detailed track record of safety violations, contributing to long-term improvements in machine operation and worker safety.

1.1 Literature Survey

[1] presents a novel approach for real-time object detection using the Single Shot Detector (SSD) method, a technique based on convolutional neural networks (CNNs). The primary objective of the study is to improve the accuracy of the SSD method for object detection tasks. The SSD method stands out for its speed and high accuracy, even when dealing with low-resolution images. One of its key advantages is the elimination of the need for a separate region proposal network, which enhances efficiency. Additionally, the SSD method has the capability to detect multiple objects simultaneously within an image, making it suitable for various applications requiring real-time object detection, such as video surveillance, autonomous vehicles, and industrial automation. However, the paper does not explicitly discuss any limitations or disadvantages of the SSD method, leaving room for further exploration and analysis in this area. [2] discusses the Open Source Computer Vision Library (OpenCV), focusing on its role in image processing and computer vision applications. It outlines key techniques like image filtering, transformation, object tracking, and feature detection. OpenCV's modules and sample applications are highlighted, showcasing its versatility in solving real-time problems. The advantages of OpenCV include being open-source, supporting multiple programming languages and platforms, and offering efficient algorithms for various computer vision tasks. The paper emphasizes continuous development and support from organizations like Intel and Willow Garage. However, it does not address potential limitations or challenges faced by users of OpenCV. [3] aims to compare the performance of three major object detection algorithms: Single Shot Detector (SSD), Faster Region-based Convolutional Neural

Networks (Faster R-CNN), and You Only Look Once (YOLOv3). The study evaluates these algorithms using the Microsoft COCO dataset and analyzes their strengths and limitations based on parameters such as accuracy, precision, and F1 score. YOLOv3 emerged as the best performer overall, offering a balance between speed and accuracy. SSD showed efficiency for real-time applications, while Faster R-CNN demonstrated the highest accuracy but with lower processing speed. The choice of algorithm is noted to depend on specific application requirements. [4] provides a comparative study of various object detection algorithms, including R-CNN, Fast R-CNN, Faster R-CNN, SSD, and YOLO v3. It aims to analyze their performance and suitability for different applications. YOLO v3 is highlighted as the most efficient and promising algorithm, offering high speed and accuracy, particularly in real-time scenarios. The study emphasizes the continuous improvement of object detection algorithms, with each iteration addressing the limitations of its predecessors. The paper concludes that YOLO v3 is a suitable choice for real-time object detection applications, given its ability to detect multiple objects in a single forward propagation. [5] explores the evolution of video surveillance systems towards networked, intelligent multi-camera systems capable of analyzing videos for object/event detection, tracking, behavior analysis, and scene understanding. Many companies are heavily investing in research and development of video analytics algorithms and intelligent surveillance products/services. Applications include multi-object tracking, event detection, scene understanding, crowd analysis, and searching video databases. It highlights the challenge posed by the overwhelming volume of raw video data generated by surveillance cameras, driving the need for intelligent video analytics to extract useful information automatically. The video surveillance market, including video analytics software, is forecasted to exceed $ 8 billion by 2010, with a fragmented value chain involving software vendors, hardware vendors, integrators, and consultants. [6] outlines a comprehensive real-time video analytics system designed for smart station surveillance, focusing on Bandung railway station. The system's objectives include implementing an automated smart CCTV surveillance system capable of real-time video analytics, employing deep learning algorithms like CNN for object detection, counting, tracking, and identification of unusual events, and utilizing big data technologies like Apache Spark and MongoDB for processing and storing large volumes of video analytics data.The system's scope encompasses real-time video analytics from multiple CCTV cameras installed at strategic locations within the railway station, such as platforms, parking areas, and commercial areas. It performs functions like object counting, tracking, and detection of dangerous objects like weapons and masked individuals. The system provides real-time alerts and notifications to authorities via email or messaging apps about detected events, enhancing security and crowd management. Advantages of the proposed system include automated real-time monitoring without continuous human supervision of CCTV feeds, improved security through the detection of threats, better crowd management through tracking passenger movement patterns, and the ability to study historical data for optimizing station operations. The system's scalability, leveraging deep learning and big data technologies, allows it to handle large volumes of video data effectively.

1.2 Motivation

The manufacturing sector in India is confronted with a significant challenge in safeguarding the well-being and safety of its workforce. Government statistics reveal a distressing trend, with an average of three workers losing their lives daily in Indian factories. Between 2017 and 2020, these incidents resulted in over 1,100 fatalities and more than 4,000 injuries reported each year. However, these figures likely underestimate the true scale of the problem. The informal structure of the economy and the tendency to underreport incidents contribute to this discrepancy. This situation underscores the urgent need for effective measures to enhance safety and security protocols in the manufacturing industry.
The implementation of video analytics for real-time monitoring offers a multifaceted approach to mitigate risks in the manufacturing sector. By harnessing the power of machine learning, artificial intelligence, image processing, and video data processing, advanced technologies like OpenCV and YOLO can be utilized to provide instantaneous analysis directly on the factory floor. This application goes beyond traditional safety measures, offering a range of benefits:

- Enhanced Safety and Security: Real-time monitoring can detect potential hazards and unsafe practices, allowing for immediate intervention to prevent accidents and injuries.
- Downtime Reduction: Predictive maintenance enabled by video analytics can anticipate equipment failures, allowing for proactive maintenance and minimizing unplanned downtime.
- Risk Minimization: By providing continuous monitoring, video analytics can identify and mitigate risks such as unauthorized access, theft, and other security breaches.
- Cost Savings: Through the prevention of accidents, reduction of downtime, and optimization of processes, video analytics can lead to significant cost savings for manufacturers.
- Remote Monitoring: Video analytics enables remote monitoring of operations, allowing for real-time decision-making and management of multiple sites from a central location.

1.3 Objectives

- Design and implement a real-time monitoring system using video analytics to detect and respond to safety and security incidents.

- Utilize machine learning algorithms, such as those available in OpenCV and YOLO, to analyze video data and identify potential safety hazards.
- Improve worker safety by providing early detection and alerts for unsafe conditions, such as equipment malfunctions, slip and fall hazards.
- Enhance operational efficiency by identifying and addressing inefficiencies in manufacturing processes through real-time monitoring and analysis.
- Facilitate remote monitoring of manufacturing facilities to enable real-time decision-making and management from a central location.
- Evaluate the performance of the real-time monitoring system in terms of its effectiveness in improving safety, security, and operational efficiency.

1.4 Problem Definition

Develop a comprehensive end-to-end machine learning system for real-time monitoring and inspection of CCTV inputs for conveyor machines to enhance operational performance and ensure workplace safety.

Chapter 2

REQUIREMENT ANALYSIS

2.1 System Model

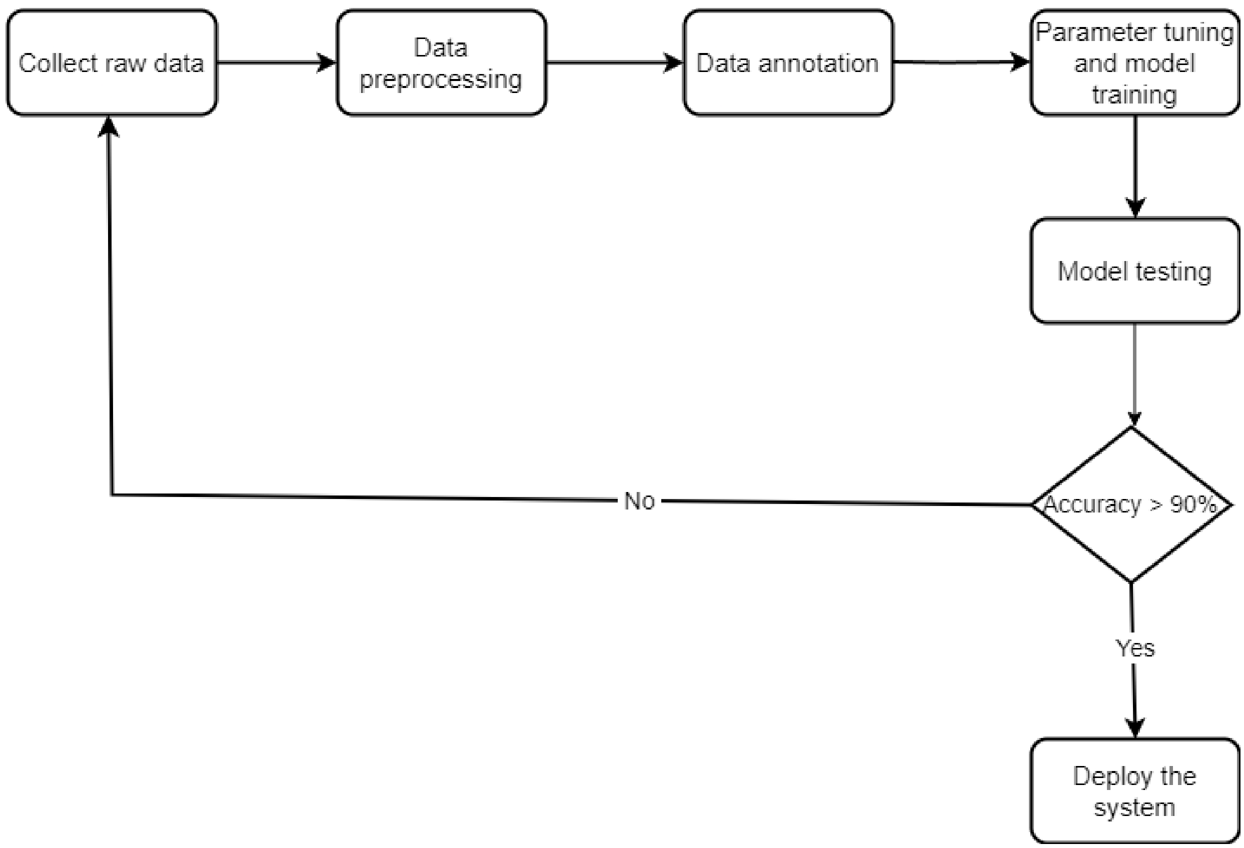

Figure 2.1: Conveyor system work flow diagram

The system model and its implementation involve a workflow designed to build an efficient video analytics system. The process began with a thorough understanding of the domain, supported by literature surveys and research to gain insights into the domain's requirements. Following this, the problem statement was analyzed to develop an approach, which included building a base object detection model. After collecting the dataset, preprocessing steps were performed to prepare the data for training. This involved tasks such as resizing the images, normalization, and augmentation to enhance the model's ability to generalize to unseen data.

The training process utilized a deep learning framework, such as TensorFlow or PyTorch, to train the object detection model. The model architecture was selected based on the requirements of the application, with popular choices including Faster R-CNN, SSD, or YOLO. Hyperparameters were tuned to optimize the model's performance. During the testing phase, the trained model was evaluated on a separate dataset to assess its performance metrics such as precision, recall, and F1 score. The model was fine-tuned based on the testing results to improve its performance further. Finally, the trained and tested model was deployed into the production environment, where it could process real-time video feeds. Monitoring and maintenance strategies were implemented to ensure the model's continued performance and to address any issues that may arise during operation.

Additionally, a custom algorithm was designed to address errors and ensure the smooth execution of the system. This algorithm was built on top of the base object detection model to enhance its functionality. Finally, the model was deployed, marking the completion of the system implementation process. The workflow involved in designing the system demonstrates a systematic approach to building a robust video analytics solution.

2.2 Functional Requirements

Functional requirements are features or functionalities that developers must include to allow users to perform their responsibilities. For the development team along with the stakeholders, it is essential to make them clear. Functional requirements often explain how a system will behave under specific circumstances.

- User shall be able to provide video or live feed as input.
- User shall be able to modify the position of reference lines.
- User shall be able to monitor missing cleats or unaligned cleats, non linear walls.
- User shall be able to modify the details that are stored in database.
- User shall be able to decide the path to store data on local system.
- User shall shall be able to receive an excel sheet containing violation details
- User shall receive alert notification through email and whatsapp.

2.3 Non-Functional Requirements

Non-functional requirements are the features of a system or product that indicate how it should behave rather than what it should accomplish. Instead of concentrating on the system or product's features and operations, they pay attention to its quality and performance. The usability, performance, reliability, security, maintainability, and other quality factors are frequently included in non-functional requirements.

1. **Performance:**
   - The system should process video frames with minimal latency, aiming for real-time performance.
   - The object detection model should have high accuracy in detecting objects of interest.
2. **Security:**
   - The system should ensure that video feeds and alert notifications are secure and not accessible to unauthorized individuals.
3. **Usability:**
   - The user interface should be intuitive and easy to use for monitoring video feeds and viewing alerts.
   - The system should provide clear and understandable alerts in case of violations.
4. **Scalability:**
   - It should be designed to easily add new object detection models or integrate with other AI/ML components.
5. **Maintainability:**
   - The code should be well-structured and documented to facilitate future maintenance and updates.
   - It should be easy to replace or upgrade components such as the object detection model or the email notification system.

2.4 Software Requirements

- Operating system: Windows/Linux/Mac 64-bit
- Programming Language: PYTHON 3.9+
- Libraries: OS, Numpy, Pandas, Tkinter, Psycopg2, cv2, YOLO, Time, Tqdm
- IDE Used: VSCode, Jupyter notebook
- Database: PostgreSQL

2.5 Hardware Requirements

- Processor: 64-bit
- Storage Space: 20 MB (excluding python libraries)
- RAM: 2 GB
- GPU: NVIDIA RTX 3060 (recommended)

Chapter 3

SYSTEM DESIGN

This chapter contains an overview of an architecture that involves several key steps to ensure systems effectiveness and efficiency. Firstly, the system requirements are analyzed to understand the scope and functionality required, such as real-time object detection, alerting, and email notifications. Next, the system architecture is designed, which includes defining the components and their interactions. For this project, components such as video processing, object detection using YOLOv8, alerting, and email notification are identified. The choice of technologies and libraries, such as Python for programming, OpenCV for video processing, and YOLOv8 for object detection, are made based on their suitability for the project requirements. The system's user interface, including any GUI elements for monitoring and configuration, are designed using python's library Tkinter. Additionally, considerations are made for performance optimization, security, scalability, and reliability. For example, hardware acceleration using GPUs may be employed to improve processing speed, while alert notifications are used for notifying about the system.

3.1 Architecture Design

YOLOv8 employs a streamlined single-head approach for predicting bounding boxes and classifying objects, incorporating an efficient anchor assignment technique called Efficient Anchor Assignment. This model utilizes a tailored multi-task loss function, combining components for bounding box regression, classification, and object confidence. To further boost its capabilities, YOLOv8 incorporates advanced data augmentation strategies, such as mosaic and copy-paste augmentation, as well as cutting-edge optimization methods like Adaptive Gradient Clipping and Adaptive Optic Weight Modulation during training. During inference, YOLOv8 applies non-maximum suppression and post-processing techniques to refine and enhance the final object detection results, delivering improved accuracy and robustness compared to its predecessors across a wide range of object detection tasks.

The architecture design of the system addresses the need for real-time and continuous monitoring of a conveyor belt to ensure the correct alignment of cleats. The system aims to detect any missing or misaligned cleats, which can adversely affect the efficiency of the workflow. To achieve this, a system was designed to raise alerts in real-time to onsite personnel. These alerts include visual alerts on a screen, hooter alarms, relay signals to halt the machine, and notifications via email and WhatsApp for remote monitoring.

The system utilizes YOLOv8, a state-of-the-art object detection algorithm, known for its accuracy and efficiency in real-time object detection. YOLOv8 employs a streamlined single-head approach for predicting bounding boxes and classifying objects, incorporating an efficient anchor assignment technique called Efficient Anchor Assignment. This model utilizes a tailored multi-task loss function, combining components for bounding box regression, classification, and object confidence. To further boost its capabilities, YOLOv8 incorporates advanced data augmentation strategies, such as mosaic and copy-paste augmentation, as well as cutting-edge optimization methods like Adaptive Gradient Clipping and Adaptive Optic Weight Modulation during training. During inference, YOLOv8 applies non-maximum suppression and post-processing techniques to refine and enhance the final object detection results, delivering improved accuracy and robustness compared to its predecessors across a wide range of object detection tasks. After training the YOLOv8 model, a weights file is obtained containing the parameters used in the detection algorithm. The workflow of the architecture involves taking a video or live feed as input, converting it into frames, and processing each frame separately. A Region of Interest (ROI) is defined for each frame, specifying the area of the frame where object detection is required. YOLOv8 object detection is then applied to the ROI, providing bounding box coordinates for the detected objects. In this case, the objects of interest are the cleats, which are either correctly aligned or misaligned. Bounding boxes are drawn around the detected objects using OpenCV, and these are displayed on the screen for live monitoring of the system. This process is repeated for each frame in the video, enabling real-time monitoring of the conveyor belt.

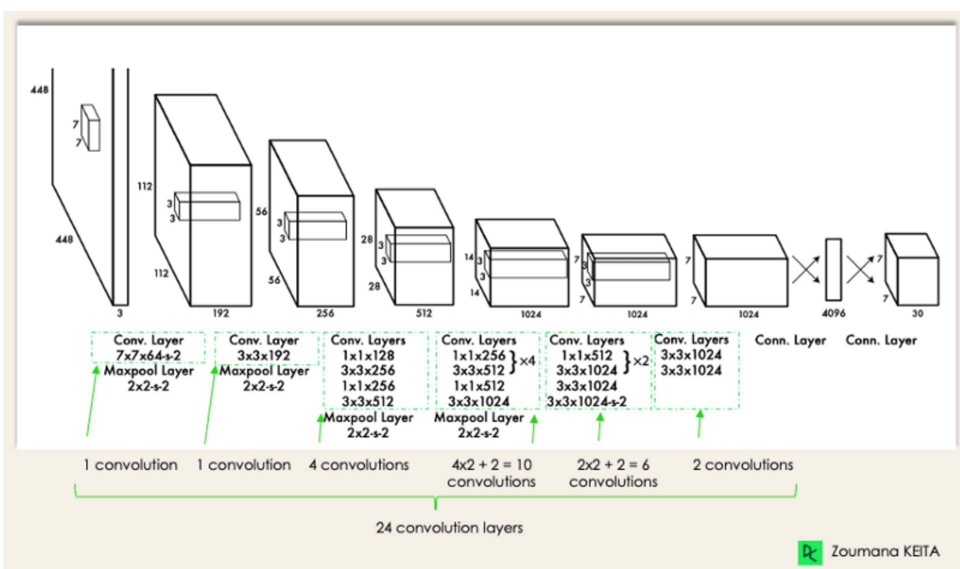

Figure 3.1: YOLO model architecture design

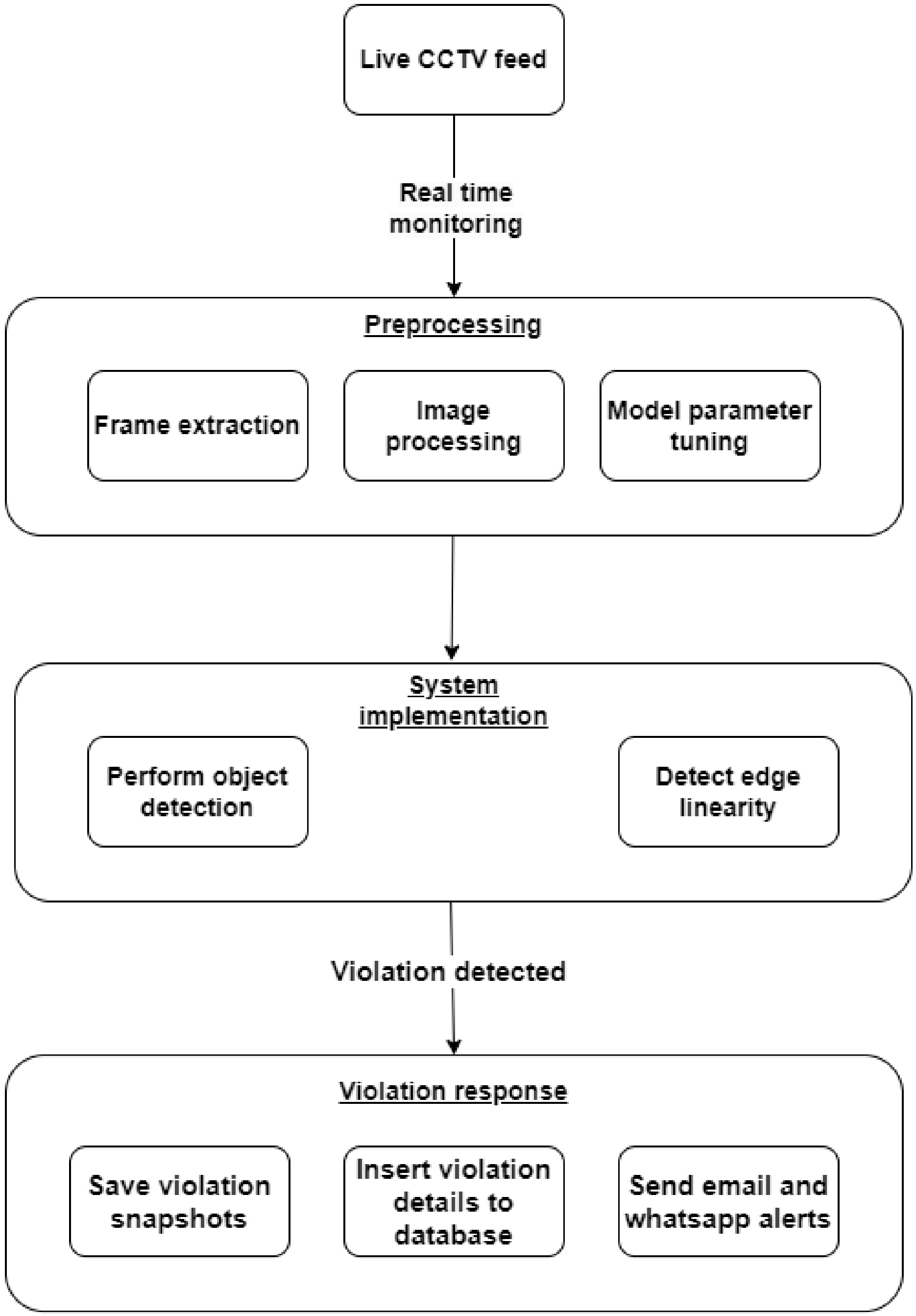

Figure 3.2: Conveyor system architecture design

The novel approach is designed for real-time monitoring and detecting safety violations in a live video feed, focusing on identifying missing cleats and non-linear walls. The detection process begins by setting up the environment and loading the pre-trained object detection model. The model weights, specifically trained to detect the required classes, are loaded to ensure accurate detection. Before starting the detection process, the video feed URL is checked to ensure it is functional and can be accessed without any issues. This step is crucial to avoid any disruptions during real-time monitoring.

The system captures each frame from the live video feed, allowing for continuous and real-time analysis of the video. Each frame is converted to gray-scale to simplify the image data and enhance edge detection. Using the Canny edge detection algorithm, edges within the frame are identified, which is particularly useful for detecting non-linear walls. Predefined reference lines are drawn on the frame to serve as a benchmark for detecting deviations, and the system checks if any detected edges intersect with these reference lines, indicating potential violations.

The model analyzes each frame to detect objects of interest, specifically looking for missing cleats and aligned cleats. When an object is detected, the model provides the coordinates of the bounding box that encloses the detected object. For live monitoring, if a missing cleat is detected, the bounding box is highlighted in red with a label indicating the type of violation. If an aligned cleat is detected, the bounding box is highlighted in green, indicating normal conditions. These visual cues are crucial for real-time monitoring, allowing operators to quickly identify and respond to potential issues. When a violation, such as a missing cleat or a non-linear wall, is detected, the system saves a snapshot of the frame locally with a timestamp and descriptive filename. The violation information, including the video source, timestamp, image name, and violation type, is stored in the PostgreSQL database for record-keeping and further analysis. Alerts can be configured to notify relevant personnel via email or messaging apps, ensuring that prompt action can be taken to address the safety violation.

3.2 Data Flow Diagram

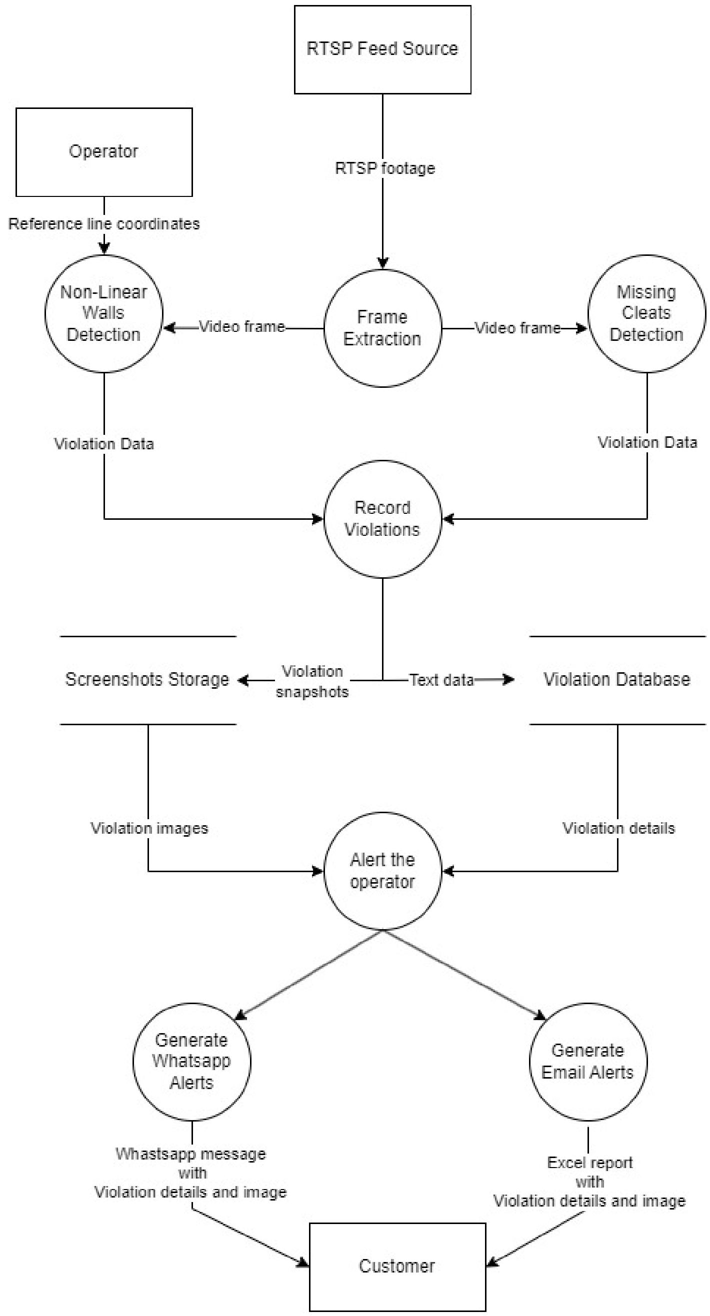

Figure 3.3: Conveyor system data flow diagram

- RTSP Feed Input: The system receives a live RTSP feed from an external camera source.
- Frame Extraction: The video feed is broken down into individual frames.
- Non-Linear Walls Detection: Each frame is processed using a Canny filter.

- Violation Detection: Detect if conveyor wall edge crosses a user-defined reference line.
- Screenshot Capture: If a violation is detected, a screenshot is taken.
- Database Update: The violation information is stored in the Violation Database.
- Missing Cleats Detection: Each frame is analyzed using a YOLO model to detect missing cleats.
- Violation Detection: If missing cleats are detected, a screenshot is taken.
- Database Update: The violation information is stored in the Violation Database.
- Generate Email Alerts: At 6am, 2pm, and 6pm, the system compiles the most recent violations into an Excel file and sends it via email.
- Generate WhatsApp Alerts: When a violation occurs, an alert is sent via WhatsApp.

3.3 User Case Diagram

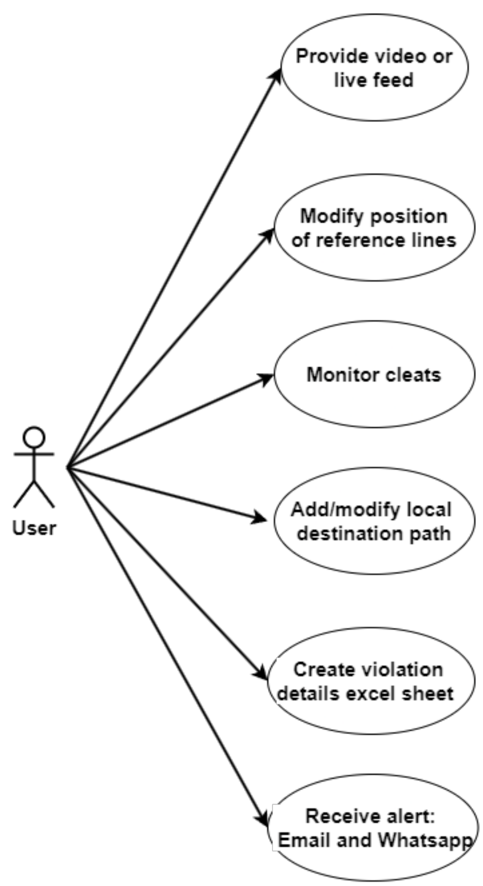

Figure 3.4: Conveyor system use case diagram

A use case diagram is a visual representation of the interactions between users (actors) and a system, showcasing the different ways the system can be used. It typically includes actors, use cases (represented by ovals), and the relationships between them (represented by lines). Use case diagrams help in understanding system requirements and serve as a blueprint for system design and development. The use case diagram for our project presents a comprehensive overview of the system's interactions with its end-users and external services. It delineates several key functionalities: Provide video or live feed, which allows users to input real-time visual data into the system; Modify position of reference lines, enabling precise adjustments for analytical accuracy; Monitor cleats, suggesting a feature for overseeing equipment or safety mechanisms; Add/modify local destination path, indicating customizable options for data storage or operational workflows; and Create violation details excel sheet, which automates the documentation of discrepancies or irregularities. Additionally, the system is designed to Receive alerts through Email and Whatsapp, ensuring prompt communication of critical updates or notifications. This diagram effectively encapsulates the user-system interaction, highlighting the project's capacity to handle complex tasks and respond dynamically to user inputs and environmental factors.

Chapter 4

IMPLEMENTATION

This chapter gives a brief description of the implementation details of the system by describing each component.

4.1 Module 1: Initialization and Setup

The "Initialization and Setup" module is the most crucial part of an end-to-end machine learning (ML) project because it lays the foundation for the entire system. Here's a detailed explanation of why this module is essential:

4.1.1 Environment Preparation:

- Library Imports: Importing necessary libraries ensures that all required functionalities are available for the project. This includes libraries for image processing (OpenCV), machine learning (YOLO), database connection (psycopg2), and others.
- Folder Creation: Storing identified photos in directories makes it easier to retrieve and review the results by carefully organizing the outputs.

4.1.2 Model Setup

Loading trained models and setting it up the by is critical as it forms the core of the detection process. Ensuring the correct model is loaded is fundamental for accurate predictions.

(model = YOLO('weights/best.pt'))

4.1.3 Video Source Configuration and Error Handling

Defining the input video source specifies the data stream that the model will process. This step ensures that the correct video feed is used for real-time monitoring. The system is also capable of processing rtsp stream. To ensure that the system can access the rtsp video feed before proceeding with further processing function CHECKRTSPWORKINGORNOT is employed.
The function CHECKRTSPWORKINGORNOT(input_name) verifies if the RTSP stream is

operational. By checking if the RTSP stream is working, the system can handle errors gracefully. If the stream is not working, the system can alert the user and prevent further execution, avoiding unnecessary processing and potential crashes. The try and except blocks in Python are used for error handling, allowing the program to manage exceptions gracefully without crashing. The try block contains the code that might raise an exception, while the except block contains the code that executes if an exception occurs. This structure helps in catching specific errors and handling them appropriately

4.2 Module 2: Real-time Notifications via WhatsApp and Email

Alerts are triggered by unaligned cleats and non-linear walls, both of which indicate potential issues that can lead to downtime and reduced efficiency of the system. Unaligned cleats, which are cleats that are not properly aligned, can result in the conveyor belt not functioning as intended, potentially causing jams or disruptions in the production process. Non-linear walls, which indicate walls that are not straight, can also lead to issues with the conveyor belt's movement, causing it to deviate from its intended path. By monitoring these aspects and generating alerts when such issues are detected, your system can proactively address them, minimizing downtime and ensuring the system operates at peak efficiency.

4.2.1 Email Alerts

A dedicated python program automates the monitoring and reporting of missing cleats in a conveyor system by prooviding email alerts. It first establishes a connection to a PostgreSQL database to store and retrieve information about missing cleats. Using the schedule library, the program sets up three scheduled tasks at specific times (6:00 AM, 2:00 PM, and 6:00 PM) to check for missing cleats and send reports accordingly. For each scheduled task, the program queries the database for records of missing cleats within the specified time range and retrieves the image names associated with them. It then creates an Excel file with details of the missing cleats, including the image of the violation and the date and time of detection, resizing the images to fit the Excel sheet. Finally, the program sends an email notification to the specified recipients (including a CC list) with the Excel file attached, containing a predefined message explaining the nature of the report and the actions needed. Overall, this program enhances safety management practices in industrial settings by streamlining the process of monitoring and reporting missing cleats.

4.2.2 Whatsapp Alerts

Another isolated pyton code is designed to continuously monitor a PostgreSQL database for unprocessed violation data related to missing cleats. It starts by reading database configuration details from a JSON file. The main function then connects to the database and fetches rows from a specific table where the status indicates the data has not been processed yet. For each fetched row, it verifies the associated image file, sends it to a cloud service using sendImagetoCould function, and sends the violation data to a specified URL using sendviloationdata function. If both operations are successful, it updates the status of the row in the database to indicate that it has been processed. This process runs in a continuous loop, ensuring that new violation data is processed promptly and efficiently.

4.3 Module 3: Non Linear Walls Detection

This is part of a larger system designed to ensure the operational efficiency and safety of machinery, where non-linear walls are critical issues that can lead to system downtime and reduced efficiency. Our unique approach is designed to monitor video frames and detect if any edges in the frames touch predefined reference lines. This ensures timely identification of potential issues and helps maintain the operational efficiency and safety of the system. If an edge touches a reference line, it indicates an alert condition which needs to be logged and saved as an image. Firstly, the system receives an RTSP live feed as input. Once this input is functioning correctly, the video is processed by analyzing each frame individually. Each frame is processed through a series of steps to better understand and evaluate the situation.

Algorithm 1 Monitoring and detecting non linear walls

Initialize:

- $current\_folder \leftarrow$ Local folder path
- Connect to Postgre database
- $rtsp\_url \leftarrow$ rtsp live feed url
- $ref\_lines \leftarrow$ define reference lines

Procedure:

while isopen($rtsp\_url$) **do**

  $frame \leftarrow$ extract_frame($rtsp\_url$)

  $grey \leftarrow$ cvtcolor($frame$)

  $edges \leftarrow$ canny($grey$)

  for $line$ `in` ref_lines **do**

    $mask \leftarrow$ zeros($edges$)

    $intersection \leftarrow$ bitwise_and($edges$,$mask$)

    $contours \leftarrow$ findcontours($intersection$)

  if length(contours)>0 **then**

    Save snapshots locally

    Upload data to postgre database

    Send email and whatsapp alerts

The novel algorithm outlines the process for monitoring and detecting non-linear walls in a live video feed using various variables to facilitate each step. Initially, the system is set up by defining the $current\_folder$ as the local folder path, connecting to the PostgreSQL database, providing the $rtsp\_url$ as the RTSP live feed URL, and establishing ref_lines as the reference lines. The procedure begins by continuously checking if the RTSP URL is open using isopen($rtsp\_url$). If it is open, the system extracts each frame using extract_frame($rtsp\_url$). Each frame undergoes a series of processing steps: it is converted to grayscale using cvtcolor($frame$) resulting in grey, and then the Canny edge detection algorithm is applied to grey resulting in edges. For each reference line in ref_lines, a mask is created using zeros(edges) and the bitwise AND operation is performed between $edges$ and the $mask$ with bitwise_and($edges$, $mask$) to find intersections, stored in intersection. Contours are identified in the intersections using findcontours($intersection$), stored in contours. If the length of contours is greater than zero, indicating that an edge has touched a reference line, the system saves snapshots locally in $current\_folder$, uploads the data to the PostgreSQL database, and sends email and WhatsApp alerts to notify of the detected non-linear walls.

This systematic approach ensures that any deviation from the expected linearity of the walls is promptly identified and reported, helping to mitigate potential issues and maintain system efficiency.

4.4 Module 4: Unalligned/Missing Cleats Detection

The detection process begins by setting up the environment and loading the pre-trained object detection model. The model weights, specifically trained to detect the required classes, are loaded to ensure accurate detection. Before starting the detection process, the video feed URL is checked to ensure it is functional and can be accessed without any issues. This step is crucial to avoid any disruptions during real-time monitoring. The model analyzes each frame to detect objects of interest, specifically looking for missing cleats and aligned cleats. When an object is detected, the model provides the coordinates of the bounding box that encloses the detected object. For live monitoring, if a missing cleat is detected, the bounding box is highlighted in red with a label indicating the type of violation. If an aligned cleat is detected, the bounding box is highlighted in green, indicating normal conditions. These visual cues are crucial for real-time monitoring, allowing operators to quickly identify and respond to potential issues.

When a violation, such as a missing cleat or a non-linear wall, is detected, the system saves a snapshot of the frame locally with a timestamp and descriptive filename. The violation information, including the video source, timestamp, image name, and violation type, is stored in the PostgreSQL database for record-keeping and further analysis.

Chapter 5

Results and Discussions

The manufacturing sector in India faces a critical challenge in ensuring the safety and well-being of its workforce, with alarming statistics highlighting the urgent need for enhanced safety measures. Video analytics offers a multifaceted approach to mitigate risks in the manufacturing sector by harnessing the power of machine learning, artificial intelligence, image processing, and video data processing. Technologies like OpenCV and YOLO enable real-time monitoring, providing instantaneous analysis directly on the factory floor. This application extends beyond traditional safety measures, offering benefits such as enhanced safety and security, reduced downtime through predictive maintenance, minimized risks such as unauthorized access and theft, significant cost savings, and remote monitoring capabilities.

Our real-time monitoring system, integrating the YOLO model, has achieved exceptional accuracy of 94% in detecting unaligned cleats, crucial for timely intervention and operational efficiency. Additionally, using OpenCV for non-linear wall detection has allowed prompt identification of deviations, preventing potential damage. The system's capability to generate emergency alerts and notifications has proven effective in mitigating risks, ensuring quick responses to incidents and reducing downtime. Seamless integration with PostgreSQL for storing violation information has facilitated detailed historical analysis and auditing, aiding continuous improvement. Maintaining a high frame processing rate, the system ensures real-time monitoring without delays, thanks to optimized algorithms and efficient processing pipelines. By detecting issues like unaligned cleats and non-linear walls in real-time, the system contributes to significant cost savings and risk minimization, enhancing safety and operational efficiency. Demonstrating scalability and flexibility, the system handles multiple video streams and various conveyor setups simultaneously, making it a versatile solution for the steel industry. User feedback has been overwhelmingly positive, highlighting the system's intuitive interface, real-time visual cues, reliability, and accuracy, reinforcing its value as a critical tool for conveyor system management.

Chapter 6

CONCLUSION AND FUTURE SCOPE

In conclusion, the proposed solution addresses a significant ongoing problem in the steel industry by providing a comprehensive system for real-time monitoring and detection of conveyor systems. Ensuring the optimal performance of these systems is crucial for maintaining operational efficiency and safety. Our solution is designed to monitor and detect unaligned cleats and non-linear walls in an industrial environment, which are critical issues affecting conveyor operations. The system begins by processing input data from live CCTV feeds, meticulously analyzing each frame to achieve the highest possible accuracy. To detect and monitor unaligned cleats, we utilized the state-of-the-art YOLO model, which predicts the coordinates of both aligned and unaligned cleats. Additionally, OpenCV was employed for image processing tasks, specifically for detecting non-linear walls.

By leveraging a range of advanced technologies, including YOLO and OpenCV, our solution provides significant benefits such as downtime reduction, risk minimization, cost savings, and remote monitoring capabilities. The system not only detects unaligned cleats and monitors non-linear walls but also raises emergency alerts to notify personnel of potential issues, thereby preventing damages and ensuring continuous monitoring of the conveyor system. Overall the proposed solution intelligently monitors the steel industry environment in real time and helps reduce downtime and save cost. Future scope includes integration with IoT devices, incorporating IoT devices can enhance the system's capabilities by providing additional data points and enabling more precise monitoring of conveyor systems. This integration can also facilitate real-time adjustments and maintenance. Also Enhanced Machine Learning Models: Future iterations of the system can integrate more advanced machine learning models and techniques to further improve detection accuracy and efficiency. This includes exploring newer versions of YOLO or other cutting-edge object detection frameworks.

Future enhancements of the project include several key areas for development and improvement:

- Integration with IoT Devices: Incorporating IoT devices can significantly enhance the system's capabilities by providing additional data points and enabling more precise monitoring of conveyor systems. This integration will facilitate real-time adjustments and maintenance, improving overall efficiency and responsiveness.

- Enhanced Machine Learning Models: Future iterations can integrate more advanced machine learning models and techniques to further improve detection accuracy and efficiency. This includes exploring newer versions of YOLO or other cutting-edge object detection frameworks, allowing for more sophisticated and reliable performance in various industrial environments.

REFERENCES

[1] K. Vaishnavi, G. Reddy, T. Reddy, N. Iyengar, and Subhani Shaik. Real-time object detection using deep learning. *Journal of Advances in Mathematics and Computer Science*, 38:24–32, 06 2023.

[2] Naveenkumar Mahamkali and Vadivel Ayyasamy. Opencv for computer vision applications. 03 2015.

[3] Divekar Chandu Anilkumar Ishika Naik Ved Kulkarni Shrey Srivastava*, Amit Vishvas and V. Pattabiraman. Comparative analysis of deep learning image detection algorithms. 03 2021.

[4] Anand John and Dr. Divyakant Meva. A comparative study of various object detection algorithms and performance analysis. *INTERNATIONAL JOURNAL OF COMPUTER SCIENCES AND ENGINEERING*, 8:158–163, 10 2020.

[5] Xueqiu Wang, Huanbing Gao, Zemeng Jia, and Zijian Li. Bl-yolov8: An improved road defect detection model based on yolov8. *Sensors*, 23(20), 2023.

[6] F Hidayat, F Hamami, I A Dahlan, S H Supangkat, A Fadillah, and A Hidayatuloh. Real time video analytics based on deep learning and big data for smart station. *Journal of Physics: Conference Series*, 1577(1):012019, jul 2020.