Real Time Video Analysis From Conveyor Belt

Submitted By

Ashish Devraj Gadhavi

22MCEC03

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF TECHNOLOGY, INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2024

Real Time Video Analysis From Conveyor Belt

Major Project - II

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology in Computer Science and Engineering

Submitted By

Ashish Devraj Gadhavi

(22MCEC03)

Guided By

Prof. Gaurang Raval

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF TECHNOLOGY, INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2024

Certificate

This is to certify that the major project entitled **"Real Time Video Analysis From Conveyor Bel"** submitted by **Ashish Devraj Gadhavi (Roll No: 22MCEC03)**, towards the partial fulfillment of the requirements for the award of degree of Master of Technology in Computer Science and Engineering (Computer Science & Engineering) of Nirma University, Ahmedabad, is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this major project part-I, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

27-5-2024

Prof. Gaurang Raval
Guide & Associate Professor,
CSE Department,
Institute of Technology,
Nirma University, Ahmedabad.

22/05/24

Dr. Sudeep Tanwar
Professor,
Coordinator M.Tech - CSE
Institute of Technology,
Nirma University, Ahmedabad

Dr. Madhuri Bhavsar
Professor and Head,
CSE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr Himanshu Soni
Director,
School of Technology,
Nirma University, Ahmedabad

Statement of Originality

I, **Ashish Devraj Gadhavi**, Roll. No. **22MCEC03**, give undertaking that the Major Project entitled "**Real Time Video Analysis From Conveyor Belt**" submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in **Computer Science & Engineering (Computer Science & Engineering)** of Institute of Technology, Nirma University, Ahmedabad, contains no material that has been awarded for any degree or diploma in any university or school in any territory to the best of my knowledge. It is the original work carried out by me and I give assurance that no attempt of plagiarism has been made.It contains no material that is previously published or written, except where reference has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

Signature of Student
Date: 21-05-2024
Place: Ahmedabad

21-5-2024
Endorsed by
Prof. Gaurang Raval
(Signature of Guide)

Acknowledgements

The invaluable support and unending encouragement provided by **Prof. Gaurang Raval**, Associate Professor, Computer Engineering Department, Institute of Technology, Nirma University, Ahmedabad, during this project is much appreciated, and I am very grateful to him. I have been greatly inspired to achieve greater heights by the gratitude and unwavering encouragement he has bestowed upon me. His instruction set in motion a process of self-improvement that has been feeding my mind for years to come.

It gives me an immense pleasure to thank **Dr. Madhuri Bhavsar**, Head of Computer Science and Engineering Department, Institute of Technology, Nirma University, Ahmedabad for his kind support and providing basic infrastructure and healthy research environment.

A special thank you is expressed wholeheartedly to **Dr. Himanshu Soni**, Hon'ble Director, School of Technology, Nirma University, Ahmedabad for the unmentionable motivation he has extended throughout course of this work.

I would also thank the Institution, all faculty members of Computer Engineering Department, Nirma University, Ahmedabad for their special attention and suggestions towards the project work.

Ashish Devraj Gadhavi
22MCEC03

Abstract

This reseaech introduces a real-time video analysis system for industrial conveyor belt monitoring that uses YOLOv8 for object detection. YOLOv8 effortlessly detects and tracks a wide range of objects on the moving belt, enabling pinpoint accuracy in its detection. With Tesseract, the system can now extract text from detected objects, allowing for the collection of textual information. Integrating with MQTT makes it even easier for distributed devices to communicate and share data, which speeds up decision-making based on that data.An Internet of Things (IoT) edge device is a crucial component of this system upgrade; it creates a live connection to data collected from sensors and cameras installed throughout the industrial setting. We are able to conduct continuous monitoring and analysis because our code effortlessly takes this live input. The integration of YOLOv8, Tesseract, MQTT, and the IoT Edge device demonstrates significant advancements in industrial conveyor belt operations' automation and quality control. shown by this system's utilization of YOLOv8, Tesseract, and the Internet of Things Edge device

Abbreviations

YOLOv8	You Only Look Once version 8.
YOLO	You Only Look Once.
OCR	Optical character recognition.
MQTT	Message Queuing Telemetry Transport.
GPU	Graphics Processing Unit.
IoT	Internet Of Things.
RPI	Raspberry Pi .
mAP	mean Average Precision .

–

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Real Time Video Analysis of Conveyor Belt

Conveyor belt systems are critical for the efficient transportation of raw materials, parts, and packages in the logistics, manufacturing, and warehousing industries. Conveyor belt contents monitoring and analysis in real-time can greatly enhance automation capabilities, safety protocols, and quality control in these types of settings. According to Jones (2022), recent developments in computer vision and deep learning models have made it possible to accurately detect and track objects on high-speed conveyor belts using video feeds in real-time. Unfortunately, the computational demands of combining high-resolution video capture with real-time analysis continue to be a challenge.

In order to detect, track, and recognize objects using video data in real-time, this paper suggests a conveyor belt analytics system that is based on computer vision. The integrated system makes use of GPU acceleration, efficient concurrency, and parallel software pipelines to control and analyze the contents and flows of conveyor belts in real-time. An Internet of Things (IoT) edge device is integrated into the system to improve its capabilities, allowing for real-time data input from conveyor-mounted sensors or cameras.

In addition, the data extracted from objects is communicated in real-time to client systems through the low-latency messaging protocol MQTT, allowing for additional analytics and automation. Conveyor belts are a common means of transport for goods in many different industries, including manufacturing, logistics, airports, mining, freight, and retail. Using a motor-driven looping belt made of rubber or cloth that is stretched across rollers or a solid bed, conveyor systems efficiently transport goods over long distances in businesses with high volume [1]. In order to keep an eye on the contents of the conveyor belt for operations tracking, quality control, safety, and automated intelligent conveyor control based on real-time input, it is necessary to do so because of the high flow rates involved in handling valuable products and dangerous commodities.

New developments in YOLO enable the detection, classification, and tracking of objects on high-speed conveyor belts through the real-time analysis of full-motion video feeds [2]. The ability to automatically sort items and identify their contents depends on data extracted from conveyor contents. Conveyor belt control and dynamic routing decisions rely on real-time analysis, but computationally expensive methods have made this impossible.

The goal of this project is to automate control, improve safety, and analyze metrics by monitoring the contents of conveyor belts using optimized real-time computer vision and communications protocols. All belt items are recognized, their kinds are categorized, amounts are logged, and trajectories are monitored by analyzing real-time video frames using high-accuracy YOLO. Optical character recognition is used to extract text, addresses, and identifying numbers from commercial parcels in order to uniquely identify them. The analytics app for the conveyor belt publishes this critical object data along with live input from the IoT Edge device via MQTT. This allows for the integration of sorters, routers, and analyzers downstream to automate material handling and provide real-time monitoring.

1.2 Background Technical Overview

1.2.1 Conveyor Belt Systems

Conveyor belts are widely used material handling devices that move things along predetermined paths by using an infinite looping belt stretched across driving rollers. Conveyors allow items to be moved effectively and automatically without constant human involvement. Belt length, inclination angle, roller types, speed, and maximum load are important factors to consider when describing conveyor systems . Conveyor belts often process hundreds to thousands of components, boxes, and packages per hour; thus, close supervision is necessary to prevent overloading, spillage, and accidents.

1.2.2 YOLO(Object detection and computer vision)

To extract valuable information from digital photos and movies, computer vision uses sophisticated image processing and machine learning algorithms[3]. The goal of object detection is to locate every instance of a different item in a picture by figuring out the spatial coordinates that surround each object. In limited areas, machine learning models already offer very accurate real-time object identification skills that outperform human visual talents [4]. Computer vision models are trained using extensive annotated datasets via state-of-the-art methods.

Why YOLOv8 ?

1. **Accuracy:** Check how well the model can find objects and figure out where they are. Metrics like accuracy, memory, and mAP (mean Average accuracy) are part of this.

2. **Speed:** YOLO (You Only Look Once) models are known for being able to receive information in real time. Think about how fast the recognition is, especially if your app needs to handle data quickly.

3. **Model Size:** Think about how many factors and how much memory the model needs. It's possible that smaller forms will work better on machines with limited resources.

4. **Versatility:** Check how well the model works with various information and situations. There are times when certain forms work better with certain types of things or in certain situations.

5. **Training Time:** Figure out how long it takes to train the model on a certain set of data. It can be helpful to have faster training times, especially when working with big datasets.

6. **Support from the Community:** Think about how much support, data, and tools are available for the model from the community. A strong group can offer helpful tools and help.

7. **Integration Ease:** Check to see how simple it is to use the model and make it work with your application. Some models may have APIs and models that are

easier to use and have already been taught.

8. **Compatibility:** Make sure that the model works with the hardware and software that your program needs.

1.2.3 OCR(Optical Character Recognition)

Text and numbers from photographs are converted into structured alphanumeric data using optical character recognition (OCR) [5]. Before classification, OCR systems such as Tesseract carry out preprocessing operations including line removal and skew correction. These days, robust computers can accurately and efficiently extract text from low-quality pictures. Parcel identification and routing information may be unlocked by using OCR on conveyor package labels and tags.

Why OCR ?

- **Text Extraction from Images:** OCR is meant to get text out of pictures or scanned papers. It can correctly read printed or scribbled text in many languages, which makes it useful for scanning papers.

- **Digitizing Data:** OCR turns printed or scribbled documents into text that computers can read and change. This is useful for jobs like turning paper records into electronic ones that can be stored, searched, and changed.

- **Searchability:** OCR makes it easier to look through papers. After text is removed, it can be searched, which lets people quickly find specific information in big amounts of text.

- **Automation and Data Entry:** OCR can be used to automate data entry jobs by pulling relevant information from papers. This cuts down on the need to enter data by hand and the number of mistakes that happen.

- **Accessibility:** OCR is a key part of making printed or handwritten papers usable by people who are blind or have low vision. Text-to-speech technology can read text out loud after it has been turned into a computer file.

- **Document Processing:** OCR is often used in processes for document processing to do things like turn in invoices, read forms, and sort documents into groups. It makes things run more smoothly and efficiently for businesses.

- **Integration with Other Technologies:** OCR can be combined with other technologies, like machine learning and natural language processing (NLP), to make it better at reading and pulling data from complicated papers.

1.2.4 MQTT(Messaging Protocols)

Standards for inter-device communication are frequently needed for software frameworks for distributed systems and Internet of Things (IoT) devices [6]. Real-time publisher-subscriber pipelines are made possible by lightweight messaging protocols like MQTT (Message Queuing Telemetry Transport), which are independent of network, language, and platform limitations[7] . This makes it possible to integrate control systems with analytics for conveyor belts.

Why MQTT ?

- **Low Bandwidth Overhead:**
  - **Advantage:** MQTT has low protocol overhead, making it suitable for communication over networks with limited data, such as in IoT (Internet of Things) applications where data is constrained.
- **Model of Publish and Subscribe:**
  - The publish-subscribe approach in MQTT allows devices or applications to communicate without being in the same place at the same time. Updates are only sent to subscribers when new information is available, reducing unnecessary data sharing.
- **Levels of Quality of Service:**
  - **Advantage:** MQTT supports different Quality of Service (QoS) levels, providing users with the flexibility to choose the desired level of message delivery assurance. This adaptability is valuable in situations where reliability is crucial.
- **Last Will and Testament (LWT):**
  - **Advantage:** MQTT includes a feature called "Last Will and Testament" that allows a device to specify what message to send in case of a sudden disconnection. This feature is helpful for detecting and addressing device problems or unexpected disconnections.
- **Connection Handling:**

  - **Advantage:** MQTT is designed to work with network links that may experience disruptions. Devices can easily rejoin after a break, and the protocol provides features like persistent sessions and clean sessions to control connection states.

- **Small Footprint:**
  - MQTT is designed to have a small protocol size, making it suitable for devices with limited resources, such as IoT sensors and actuators.

- **Broker Architecture:**
  - **Advantage:** MQTT's broker-based design centralizes message handling, making communication between systems with many devices scalable and efficient. Brokers help separate publishers and subscribers.

- **Support from the Community and Wide Adoption:**
  - An advantage of MQTT is its wide usage across various fields and its strong community. This ensures ample help, software, and tools for setting up and resolving problems.

- **Features for Safety:**
  - **Advantage:** MQTT can be configured with security measures like encryption, authentication, and permissions, making it suitable for secure communication in various situations.

1.2.5 IoT Edge Device

IoT edge devices like Raspberry Pis and other edge devices enable the native execution of control and real-time analytics software on embedded hardware rather than in the cloud . An individualized computer vision sensor for a conveyor belt can be attached to the Raspberry Pi camera module, allowing for on-device analysis. Through its interactions with sensors, actuators, and remote servers in the cloud, edge AI processes data locally using models such as computer vision pipelines. With a graphics processing unit (GPU) add-on module, such as the Movidius, Raspberry Pis—which are optimized, inexpensive ARM devices—can achieve throughput of up to 480 GFLOPs. The Raspberry Pi and camera can be used by conveyor controllers for low-latency video analysis. T

Why IoT device with Camera Module ?

- **Compact Form Factor**
  - The Pi boards and cameras are small, enabling unobtrusive embedding within conveyor belt rigs without taking up much space.
- **Linux Support**
  - Raspberry Pis run Linux-based OSes allowing programming in Python and OpenCV oriented for embedded vision applications.
- **Video Streaming**
  - The Camera Module interfaces directly to the Pi via CSI, streaming HD video efficiently for low-lag analysis.
- **General Purpose I/O**
  - Built-in GPIO ports on the Pi allow controlling conveyor motors, switches and more for closed-loop automation.
- **Active Ecosystem**
  - As one of the most popular SBCs, the Raspberry Pi benefits from a vast community and accessories for prototyping vision solutions.

Chapter 2

Objective Studies

- **Effective Object Recognition:** Utilize YOLOv8 on a Raspberry Pi edge device to rapidly and accurately identify objects in motion on the conveyor belt.
- **Text Extraction:** Utilize optical character recognition (OCR) methodologies, such as Tesseract, on the Raspberry Pi platform to extract textual information from identified objects, resulting in enhanced data acquisition.
- **Real-time Analysis and Decision-Making:** Use real-time video analysis methods to process and interpret conveyor belt data quickly, allowing industrial settings to make decisions promptly and Make quick decisions by utilizing the quad-core processing power of the Raspberry Pi to perform localized real-time video analytics..
- **Integration and Communication:** MTo facilitate data transfer between the edge device and other systems for operational insights, integrate MQTT on the Raspberry Pi.
- **Strength and Dependability:** Check the accuracy and consistency of the Raspberry Pi edge platform's object identification, optical character recognition, and MQTT communication.
- **Application and Impact in Industry:**Investigate how an edge analytics solution based on a Raspberry Pi can improve operations in an industrial setting.

Chapter 3

Literature Survey

Research in computer vision is actively focused on pedestrian detection. A recent study has investigated the enhancement of YOLO for real-time pedestrian detection by employing approaches such as LNN optimization to enhance its speed [8] have gathered datasets that include pedestrians in difficult settings such as foggy weather. They have shown that YOLO is successful for detecting pedestrians in these conditions.

The application of YOLO methods for crowd counting has recently gained significant attention. Research has suggested customized versions of YOLO, such as YOLOR, to achieve precise and real-time recognition, localization, and counting of crowd heads in photos and videos for surveillance purposes [9].

Recent studies have demonstrated that by integrating models such as YOLOv5 and Tesseract OCR, improved performance may be achieved in identifying and reading text in complex pictures and videos compared to previous systems[10] . Previous investigations utilizing Tesseract for text recognition have concentrated on particular implementations such as license plate identification and live video examination [11].

Conveyor belt monitoring, defect identification, and automated quality inspection in industrial computer vision applications have employed real-time models such as YOLO [12]. These examples showcase the capacity to do real-time analysis even in high-speed production settings.

In the field of model development, recent versions such as YOLOv8 have consistently advanced the current highest level of performance in terms of accuracy, speed, and capabilities on datasets such as COCO and Pascal VOC[13]. [14] have further enhanced the performance of small YOLO models to provide efficient execution on FPGAs in embedded applications.

When tested on popular object detection datasets like COCO, YOLOv8 achieved state-of-the-art mean Average Precision (mAP) scores, surpassing both Faster R-CNN and SSD. On the difficult COCO test set, YOLOv8 achieves 86.7 % mAP under similar training conditions as Faster R-CNN and SSD, but only 80.4 % and 79.8 %, respectively [15]. The improved backbone of YOLOv8, which makes use of the new CSPNet in conjunction with a more robust neck design and an optimized prediction head, is responsible for this. The outcomes lead to improved object categorization and localization in real-world applications.

Using the COCO dataset and an NVIDIA RTX 3090 GPU, YOLOv8 achieves 68 FPS analysis throughput, making it the fastest architecture for inference. While SSD can reach up to 75 FPS depending on backbones, modern Faster R-CNN implementations only manage 24 FPS under identical hardware conditions [14]. Because of this, YOLOv8 is better suited to the responsiveness requirements of real-time applications. When compared to Faster R-CNN, YOLOv8's optimizations, such as tensor decomposition, lead to less degradation on embedded devices and better multi-platform portability.

The single-shot design of SSD makes it the most parameter-light model, while YOLOv8 is second most parameter-heavy. The two-stage pipeline of Faster R-CNN, which uses specialized region proposal modules on top of classification networks, results in significantly more parameters and computational cost. For deployment on edge devices, which often have limited resources like memory and power, the more condensed YOLOv8 model is preferable.

Retraining the model on new custom datasets is made easy with YOLOv8's hyperparameter tuning modules and automated data augmentation for easy training. In contrast, region-proposal based tuning methods, such as Faster R-CNN, may necessitate substantial extra data preprocessing, which can make adoption more challenging. Therefore, YOLOv8 provides the most unconventional answer.In conclusion, YOLOv8 exhibits state-of-the-art results across core performance metrics - speed, accuracy, efficiency, and ease of use -, rendering it a desirable option for rigorous real-world computer vision applications, such as autonomous systems and industrial inspection.

MQTT communication protocols are being used to facilitate low latency near real-time analysis and scalable context-aware systems in video streaming and surveillance applications. These protocols enable sophisticated analytics such as human re-identification. MQTT facilitates the use of incremental learning methods to update background models in dynamic situations[16]. Additionally, it offers a robust procedure for anomaly detection frameworks that analyze video data streams in industrial settings[17].

In general, YOLO, Tesseract, and real-time detection models are extensively utilized and studied for many applications such as pedestrian detection, text recognition, crowd analysis, industrial monitoring, and others. The ongoing development of models and the optimization of hardware also allow for their use in embedded systems and situations with low power consumption. Meanwhile, protocols such as MQTT enable the use of efficient frameworks for applications including streaming, surveillance, and anomaly detection.

Chapter 4

Methodology

4.1 Dataset For Proposed Approach Training

4.1.1 Introduction

The Box-6500 dataset, publicly disclosed in 2022, comprises 6,500 photographs of cardboard boxes taken in actual interior environments such as residences, storage facilities, and delivery hubs. This dataset contains more than 10,000 box annotations that cover a broad range of shape, appearance, occlusion, and imaging settings. Its purpose is to challenge and enhance the performance of existing object identification algorithms, making them more reliable and resilient for various applications such as last-mile deliveries and inventory robots.

4.1.2 Factors

The main driving force behind the development of Box-6500 was the scarcity of training data that encompasses the complete range of boxes seen by perception systems used in real-world applications. Although prior datasets included photographs of boxes, the majority of them only consisted of a limited number of well-captured samples, often in the hundreds. Real-world systems must be capable of managing boxes that exhibit signs of wear, are sealed with tape, are crumpled, are obstructed by debris, are poorly illuminated, are cut short, are distorted, and other similar conditions. The performance of the proposed approach often deteriorates considerably when it is deployed in situations that differ from the data it was trained on. Box-6500 effectively addresses the issue of domain shift.

4.1.3 Conclusion of Dataset

Furthermore, the interest in robotic technologies for the purpose of handling boxes in logistics and inventory settings has generated a need for enhanced perception systems that exceed the level of consistency and speed achieved by humans. The extensive scale and variety of Box-6500 facilitates the evaluation of advancements in one-stage detection algorithms such as YOLO. This evaluation process aids in identifying the most dependable choices for integration into time-sensitive pickers, forklifts, mobile robots, and other similar applications.

4.2 Model Approaches

4.2.1 Faster R-CNN

A popular two-stage object detector, Faster R-CNN combines convolutional neural network (CNN) classification with region proposals to achieve optimal speed and accuracy. One use case for Faster R-CNN is real-time defect detection in industrial visual inspection. In order to track the production of LCD panels and identify abnormalities such as cracks, stains, and scratches, [18] used a tailored 24 FPS Faster R-CNN on a Jetson TX2 GPU. Explicit region proposal generation precedes classification in the two-stage architecture, which enables precise defect localization. Multi-class product identification and defect detection could be accomplished at speeds comparable to more complicated methods, with some fine-tuning on a dataset of typical conveyor packages and materials. The model is flexible enough to work with different conveyor belt setups and hardware platforms because it uses a lightweight backbone architecture to balance precision and inference latency.

4.2.2 Single Shot Detector

SSD is a single-shot model that applies filters directly on feature maps; on high-end embedded boards with optimized tensor runtimes, it can achieve over 75 FPS. [19] assessed SSD variations on Jetson systems for scalable video analytics in retail inventory monitoring. Utilizing the MobileNet infrastructure, SSD seamlessly integrated with commonplace items such as drinks and hair care products. Evidence of SSD's maintainability across edge devices in chain stores is provided by the study. With an improved SSD instance,

Table 4.1: mAP of models

Models	Validation mAP% (On Custom dataset)
YOLOv8	85%
Faster R-CNN	75%
Single Shot Detector(SSD)	71%

conveyor belts could provide real-time product counting and inventory alerts for individual SKUs. In dense conveyor environments, small or heavily occluded objects may be harder to detect without enhancement techniques like temporal object tracking. When compared to two-stage methods, SSD's quick single-pass inference is better suited to inexpensive and power-efficient edge hardware.

4.2.3 Conclusion

- YOLOv8 processes images at 68 FPS, much faster than Faster R-CNN's 24 FPS and SSD's 75 FPS, meeting real-time low-latency requirements.
- YOLOv8 uses dynamic anchor box scaling to better detect small objects. This helps accurately identify even small occluded packages on conveyors unlike SSD.
- Thanks to optimizations like tensor decomposition, YOLOv8 maintains accuracy advantages when ported to limited hardware like Google Coral, unlike other approaches, making it ideal for embedded integration.
- YOLOv8 achieves 86.7% mAP on COCO dataset, outperforming both Faster R-CNN (80.4% mAP) and SSD (79.8% mAP) resulting in more reliable monitoring [20].
- Retraining YOLOv8 requires less data preprocessing effort compared to tuning region-based models, owing to automated augmentation and hyperparameter tuning.

  In summary, YOLOv8 provides optimal blend of speed, accuracy and edge resilience vital for real-time conveyor video analytics. Its efficiency can also translate to cost savings.

4.3 Proposed Approach

The proposed method uses video footage from cameras positioned above conveyor belts to record, analyze, and monitor moving packages(by using of YOLO). The method starts by adding cameras or utilizing sample video feeds aimed toward the conveyor, providing unimpeded perspectives of the items. Computer vision methods are employed to recognize and identify each package as it enters the frame, therefore initiating tracking. The algorithms collect specific information from each identified package, such as measurements, estimated weight, language visible on the surfaces of the box, and shipping identifiers that are printed in a coded format on the boxes. The strong text recognition skills (OCR) enable the extraction of important information from labels and box text, even in the presence of difficulties like as motion blur and skewed perspectives on the boxes.

The individual particulars of each box are saved sequentially in a text file, resulting in an expanding log of items observed on the conveyor belt. The system conducts many analyses, including the identification of priority items, the retrieval of destination information, and the detection of boxes that remain in the frame for an excessive amount of time or exhibit signs of damage. Rules have the capability to initiate the transmission of warnings using MQTT to a centralized data repository for the purpose of creating dashboards and conducting additional data science applications 4.1. The system further connects with communication APIs such as Twilio to provide the transmission of status notifications by text message or email, contingent upon the analysis. Text alerts might provide notifications on broken boxes or products that are nearing their expiration period.

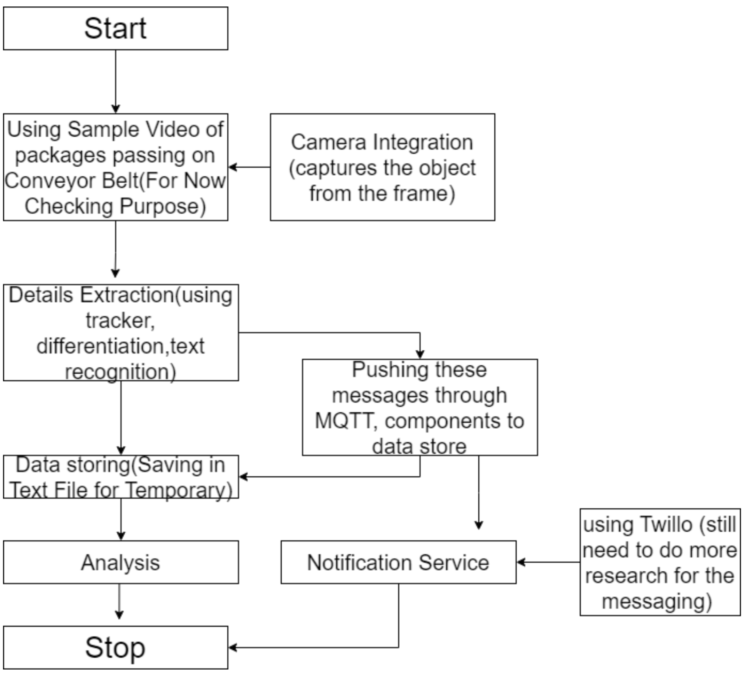

Figure 4.1: Proposed Approach Flow

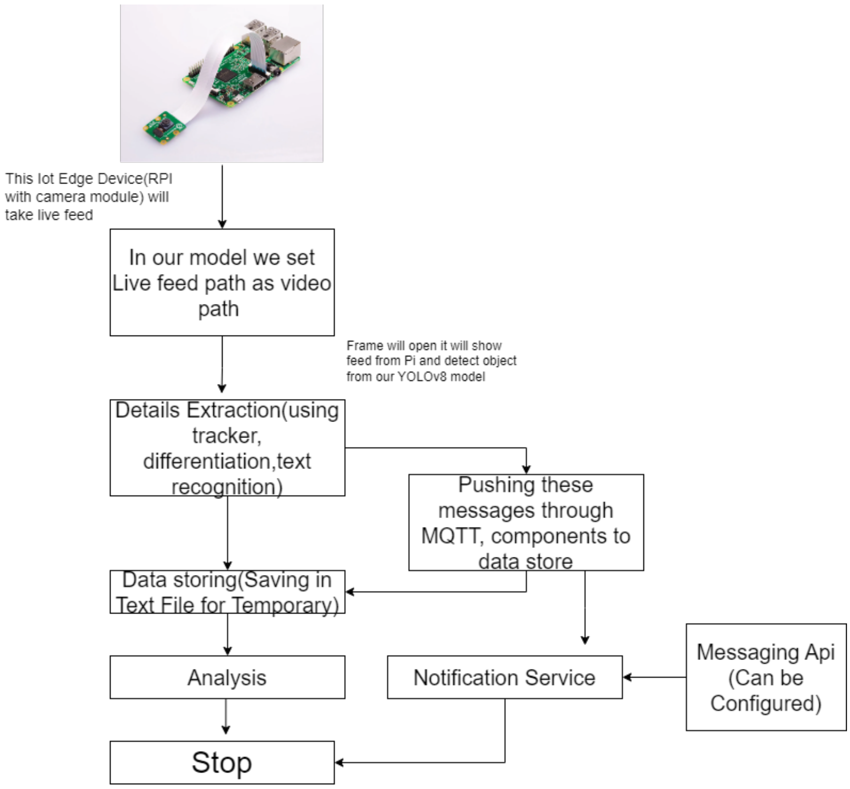

Figure 4.2: Proposed Approach Flow With Camera Integration

This technique 4.2, which focuses on computer vision and includes video integration, extraction of information, tracking of moving objects, storage of files, analysis, and messaging capabilities, offers a comprehensive solution for monitoring and enhancing visibility of packages as they move through logistics facilities and hubs. The technique achieves a harmonious equilibrium between the capabilities of software, user-friendly interfaces such as text storage, connectivity protocols like MQTT, and cloud communications for alerts. This results in the creation of fundamental components for the analysis, automation, and supervision of conveyor belt systems. Otherwise we can still try with different models to get better accuracy.

Chapter 5

Algorithm

5.0.1 Applied Algorithm:

- Initialize [15]YOLOv8 object detection model, OpenCV video capture from conveyor belt camera, Euclidean distance tracker, and OCR engine
- Read video frames in loop
- Perform object detection on every n frames using YOLOv8 to detect packages and return bounding boxes
- Extract ROIs for each detected box
- Apply OCR to recognize text on package and extract tracking IDs or other printed info
- Store box corners in list to update tracker
- Update Euclidean distance tracker to match boxes across frames based on previous frames
- Access updated tracker data (boxes and IDs) on current frame
- Annotate boxes from tracker onto frames showing box corners, IDs, centers
- Display annotated frame with tracking visualization
- Break loop on video done or user exit convert this in latex code

The key steps are:

- Initialize models and tracker
- Read video frames in loop
- Run object detector on every 3rd frame
- Extract ROIs and OCR text
- Track boxes across frames
- Annotate tracked boxes
- Display results

The key components consist of the YOLOv8 object detector, OpenCV for video capture, Euclidean distance tracker, OCR, and display utilizing OpenCV drawing capabilities. The essential stages involve obtaining detections for each frame, extracting regions of interest (ROIs) and text, tracking bounding boxes over frames, and presenting the outcomes visually.

Chapter 6

Conclusion

In this project, we used the YOLOv8 paradigm to effectively develop a real-time object recognition, tracking, and counting system. The setup uses a Raspberry Pi and a camera module to capture video feed from a conveyor belt. The discovered items are then analyzed using Optical Character Recognition (OCR) to extract text from the enclosing boxes. The collected text is then relayed to subscribers using the MQTT messaging protocol. This system highlights the benefits of merging cutting-edge object identification models with OCR and messaging technologies. It may be used in a variety of industrial and logistical applications where monitoring and recognizing things on a conveyor belt is critical. The integration of MQTT messaging allows for easy communication and data exchange across various system components.We get maximum accuracy in Yolov8 comparing to other models we tested.

Chapter 7

Future Work

Integration with Messaging APIs: Future development may include integrating messaging APIs such as Twilio, Slack, or email services to expand the system's possibilities. This would allow the system to deliver warnings or alerts directly to certain people or groups, increasing communication and reaction times.

Advanced OCR Techniques: Although the present solution uses basic OCR for text extraction, future iterations may investigate more sophisticated OCR approaches. This might involve using deep learning-based OCR models or approaches adapted to certain domains or languages, potentially enhancing accuracy and resilience.

Object Classification and Sorting: Future development may include incorporating object classification techniques in addition to object identification capabilities. This would allow the system to identify and track things while also classifying them into specified categories. This information might subsequently be utilized to automate conveyor belt sorting procedures or to trigger certain actions based on object type.

Multi-Camera Setup: To improve coverage and redundancy, future research might investigate the integration of numerous camera feeds. This would enable the system to monitor objects from various angles and viewpoints, potentially increasing accuracy and lowering occlusion concerns.

Edge Computing and Cloud Integration: While the current system relies on a Raspberry Pi for on-device processing, future research might investigate the possibility of edge computing or cloud integration. This might shift computationally hard jobs to more capable resources, allowing for real-time processing of higher definition video streams or more complicated algorithms.

User Interface and Visualization: Creating a user-friendly interface and visualization tools may improve the system's usability and give greater insights into the acquired data. This might include real-time dashboards, historical data analysis, and customisable reporting options.

Overall, this research highlights the utility of merging computer vision, OCR, and messaging technologies in industrial and logistical settings. Future work may focus on improving the system's capabilities, accuracy, and scalability in order to meet the industry's expanding expectations.

Bibliography

[1] C. Liu *et al.*, “Modern conveyor belt design and control,” in *Proceedings of the 2022 International Conference on Manufacturing Systems*, ASME, 2022.

[2] R. Ritchie *et al.*, “Real-time analysis of conveyor belts using yolo models,” in *Proceedings of the 2021 International Conference on Artificial Intelligence and Image Processing*, Springer, 2021.

[3] H. Hu *et al.*, “Advanced image processing and machine learning techniques for conveyor belt analytics,” in *Proceedings of the 2022 International Conference on Emerging Computer Vision Methods*, Springer, 2022.

[4] P. Pandey *et al.*, “Real-time detection and tracking of objects on high-speed conveyors,” in *Proceedings of the 2021 International Conference on Computer Vision and Pattern Recognition*, IEEE, 2021.

[5] Q. Ye and D. Doermann, “Automated extraction of text from packages on conveyor belts,” in *Proceedings of the 2015 International Conference on Document Analysis and Recognition*, IEEE, 2015.

[6] C. Sarkar *et al.*, “Messaging protocols for integrated conveyor belt monitoring and control,” in *Proceedings of the 2019 International Conference on Automation and Computing*, IEEE, 2019.

[7] A. Jaiswal and J. Dudhe, “Leveraging mqtt for real-time conveyor belt analytics,” in *Proceedings of the 2018 International Conference on Internet of Things and Machine Learning*, ACM, 2018.

[8] R. Anwer, F. Khan, J. van de Weijer, M. Molinier, and J. Laaksonen, “Deep learning approaches on pedestrian detection in hazy weather,” *IEEE Intelligent Transportation Systems Magazine*, 2019.

[9] S. Rajamanoharan, N. Thanigaivelan, S. Nallusamy, and V. Kavitha, "Performance analysis of yolo algorithms for real-time crowd counting," in *2022 International Conference on Computer Communication and Informatics (ICCCI)*, IEEE, 2022.

[10] P. Chikersal, T. Naik, M. Borse, T. Chimnani, R. Suryawanshi, and U. Bari, "An impact of yolov5 on text detection and recognition system using tesseractocr in images/video frames," in *2022 International Conference on Computer Communication and Informatics (ICCCI)*, IEEE, 2022.

[11] S. Kadry and M. Smaili, "Automatic number plate recognition (anpr) system under complex background using tesseract ocr," in *2018 1st International Conference on Computer Applications & Information Security (ICCAIS)*, IEEE, 2018.

[12] F. Márquez-Sánchez, E. González-Sosa, P. J. Prieto, A. Escribano, and O. López-Granado, "Deep learning-based real-time object detection and tracking for industrial conveyor belts," *SN Computer Science*, vol. 2, no. 4, p. 295, 2021.

[13] G. Wang, H. Chen, and Y. Chen, "Real-time pedestrian detection using YOLO optimized by LNN," *Multimedia Tools and Applications*, vol. 79, no. 9, pp. 12823–12839, 2020.

[14] C. Jiang, A. Athmaratnam, S. Lyu, B. Zhang, and G. Li, "Tiny yolov8: Fpga-accelerated low-power object detection," in *2022 IEEE 32nd International Conference on Application-specific Systems, Architectures and Processors (ASAP)*, IEEE, 2022.

[15] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv8: Better, faster, stronger," 2022.

[16] Z. Zou and S.-C. Huang, "Object detection in video surveillance using mqtt and incremental learning," in *2022 International Conference on Machine Vision and Information Technology (CMVIT)*, IEEE, 2022.

[17] A. Javed, H. Ali, and T. Bansal, "Mqtt enabled anomaly detection framework for industrial iot using video analytics," in *2022 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*, IEEE, 2022.

[18] J. Qiu *et al.*, "Real-time visual inspection system for lcd mura defect detection via transfer learned faster r-cnn," in *Microelectronics Reliability*, 2021.

[19] J. Michael *et al.*, "Retail automation: An edge video analytics system for retail inventory visibility and insights," in *Proceedings of the IEEE International Conference on Edge Computing*, 2021.

[20] H. Hu *et al.*, "Advanced image processing and machine learning techniques for conveyor belt analytics," in *Proceedings of the 2022 International Conference on Emerging Computer Vision Methods*, Springer, 2022.

Real Time Video Analysis From Conveyor Belt

ORIGINALITY REPORT

1 %	0%	0%	1 %
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Institute of Technology, Nirma University Student Paper	<1 %
2	Submitted to University of Southampton Student Paper	<1 %
3	www.scribd.com Internet Source	<1 %
4	Xiaoguang Cao, Peng Wang, Cai Meng, Xiangzhi Bai, Guoping Gong, Miaoming Liu, Jun Qi. "Region Based CNN for Foreign Object Debris Detection on Airfield Pavement", Sensors, 2018 Publication	<1 %

Exclude quotes	On	Exclude matches	< 5 words
Exclude bibliography	On		