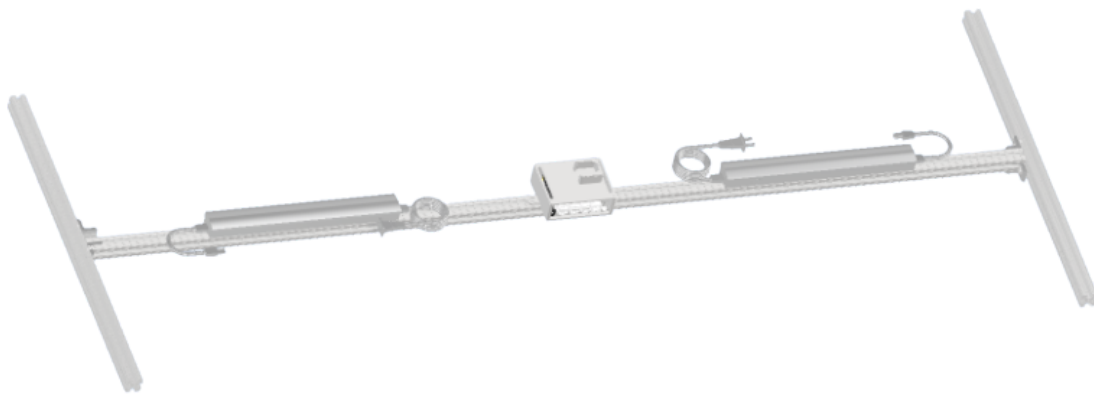


Conveyor Belt Monitoring System Manual

Automated Maintenance Inspection for Amazon Fulfillment Centers



Date Updated: 4/29/23

Prepared for: Amazon RME [MQJ5]

Created by: Team 5 ~ Purdue University [ENGT 481]

Introduction	3
1.1. Purpose of the manual	3
1.2. Scope of the web app	3
1.3. Overview of the web app	3
1.4 Automated Scheduling	3
1.5. Terminology and definitions	4
System Requirements	4
2.1. Hardware requirements	4
2.2. Software requirements	4
2.3. Supported browsers	5
Installation and Setup	5
3.1. Installation instructions	5
3.2. Configuration instructions	6
3.3. Licensing information	7
Fixture and Hardware Mechanical Designs	7
4.1. Full Assembly - Fixture & Nvidia Jetson Case	7
4.2. Nvidia Jetson Case - Assembly	8
4.3. Nvidia Jetson Case - Jetson	8
4.4. Nvidia Jetson Case - Camera	9
4.5. Nvidia Jetson case - Mount Lid	9
User Interface	10
5.1. Overview of the UI	10
5.2. Navigation and menus	10
5.3. Dashboard and widgets	10
5.4. Forms and input controls	10
5.5. Reports and data visualization	10
5.6. Index.html	10
5.7. Results.html	11
5.8. Data.html	12
Data Management	12
6.1. Data sources and integration	12
6.2. Data import and export	13
6.3. Data validation and cleansing	13
Functionality	13
7.1. Features and modules	13
7.2. Use cases and scenarios	13
7.3. Business logic and rules	14
7.4. Scheduled processing	14
Maintenance and Troubleshooting	14
8.1. System Maintenance Tasks	14
8.2. Performance Tuning and Optimization	15
8.3. Error Messages and Logs	15
8.4. Known Issues and Limitations	15
Appendix	16
9.1. Glossary	16
9.2. References, Resources, FAQ	17

Introduction

The Conveyor Belt Monitoring System is an application that monitors the condition of a conveyor belt used at Amazon Fulfillment Centers. The Conveyor Belt Monitoring System is a web application designed to automate the process of monitoring the condition of the conveyor belts. This system uses computer vision to analyze images captured by a camera mounted above the conveyor belt to extract key features such as the straightness of the belt, the number of rips on the belt, and the number of vertices on the belt surface. Based on these features, the system generates a report indicating the condition of the belt and whether maintenance is required.

1.1. Purpose of the manual

The purpose of this manual is to provide instructions on how to use the Conveyor Belt Monitoring System. It includes information on the requirements needed to run the code, how to run the code, and how to interpret the results generated by the system. This manual is intended for users who want to monitor the condition of the conveyor belts in their manufacturing or logistics facility.

1.2. Scope of the web app

The scope of the web application is to automate the process of monitoring the condition of conveyor belts used in manufacturing or logistics facilities. The system uses computer vision to extract key features of the belt, such as its straightness, the number of rips, and the number of vertices on its surface. Based on these features, the system generates a report that indicates whether maintenance is required for the belt. The web application allows users to initiate the inspection process, view the inspection results, and view the historical data of previous inspections.

1.3. Overview of the web app

The Conveyor Belt Monitoring System is a web application that uses computer vision to monitor the condition of conveyor belts. The system captures images of the conveyor belt using a camera mounted above the belt and processes these images using OpenCV functions. The system then extracts key features of the belt, such as its straightness, the number of rips, and the number of vertices on its surface. Based on these features, the system generates a report that indicates whether maintenance is required for the belt. The web application allows users to initiate the inspection process, view the inspection results, and view the historical data of previous inspections.

1.4 Automated Scheduling

In addition to manual triggering of the conveyor belt monitoring system, the system includes automated scheduling functionality using the APScheduler library. The system triggers the /process, /results, and /data routes automatically every day at 10am using the APScheduler. This ensures that the system runs regularly without requiring manual intervention.

To implement automated scheduling, the system initializes the APScheduler and adds a job to trigger the /process, /results, and /data routes at 10am every day using the 'cron' trigger. The default values for speed and length are set to the benchmark values, but these can be adjusted based on the specific requirements of the facility. The scheduler is started and runs continuously in the background, ensuring that the system is always running as expected.

1.5. Terminology and definitions

- Computer Vision: A field of study that focuses on enabling machines to interpret and understand visual data from the world around them.
- OpenCV: An open-source computer vision and machine learning software library that provides a wide range of image processing functions.
- Flask: A lightweight web application framework that provides tools for building web applications quickly and easily.
- SQLite: A lightweight relational database management system that is included with Python.
- Feature: A specific characteristic or property of the conveyor belt that is extracted by the system, such as the straightness of the belt or the number of rips on the belt.
- Maintenance: The process of repairing or replacing worn or damaged parts of the conveyor belt to ensure its proper functioning.

System Requirements

The Conveyor Belt Monitoring System has specific hardware and software requirements that must be met in order for it to function correctly. This section outlines the necessary requirements for running the system on our current device (Nvidia Jetson Nano) with the 1080p 60fps USB camera are listed below, but can be changed to any other hardware.

2.1. Hardware requirements

- Nvidia Jetson Nano Developer Kit or equivalent
- 1080p 60fps USB camera
- Stable internet connection (with a static IP, if accessing from multiple devices)

2.2. Software requirements

- Ubuntu 18.04 , Windows 10, or MacOS 10.9 or higher
- Python 3.6 or higher
- Flask framework
- OpenCV 4.5 or higher

- NumPy 1.19 or higher

2.3. Supported browsers

- Google Chrome (version 80 or higher)
- Mozilla Firefox (version 75 or higher)
- Apple Safari (version 13 or higher)

Note: It is recommended to use the latest version of the above browsers for the best performance and user experience. Other browsers may not be fully compatible with the web application.

Before running the system, ensure that all the above hardware and software requirements are met. Failure to meet these requirements may result in the system not functioning as intended or not working.

Installation and Setup

3.1. Installation instructions

/home/dh/Desktop/Belt-System-Final/

```
|— requirements.txt
|— results.db
|— run.py
|— Edge/
|— Frame/
|— Rips/
|— Surface_defects/
|— template_folder/
| |— results.html
| |— data.html
| |— index.html
| |— navbar.html
|— app.py
```

Part 1, your computer (**not recommended**) To install and set up the Conveyor Belt Monitoring System on my provided device, please follow the steps below

1. On your computer (if you want to connect remotely)
 - a. `ssh dh@10.165.101.242` (find it's current ip address via `ifconfig` in terminal)
 - b. password: purdue

c. then in the terminal:

```
cd /home/dh/Desktop/Belt-system-final
source myenv/bin/activate
python run.py
```

Part 2 (recommend) on the device:

2. Or it would be easier to go on the Jetson Nano, open terminal, copy & paste this code:

```
cd /home/dh/Desktop/Belt-system-final
source myenv/bin/activate
python run.py
```

To install and set up the Conveyor Belt Monitoring System on a new device, please follow the steps below:

1. Clone the repository from the following GitHub link: <http://github.com/wyniemko/prod-amazon-belt>
2. Install Python 3.6 or higher on your system. (pip _____ 'library name' ex. Flask, NumPy)
3. Install the Flask framework, OpenCV 4.5 or higher, NumPy 1.19 or higher, and all other packages using pip. You can do this by running the following command in your terminal or command prompt:

```
pip install -r requirements.txt
```

4. Connect a 1080p 60fps USB camera to your system.
5. Navigate to the directory where the code is located in your terminal or command prompt.
6. Run the following command to start the web application:

```
python run.py
```

7. Open a web browser and navigate to <http://localhost:5000/> to access the home page of the web application.

3.2. Configuration instructions

The Conveyor Belt Monitoring System does not require any configuration before use. However, if you want to change the port number used by the web application, you can do so by modifying the following line in the run.py file:

```
app.run(host='0.0.0.0', port=5000, debug=True)
```

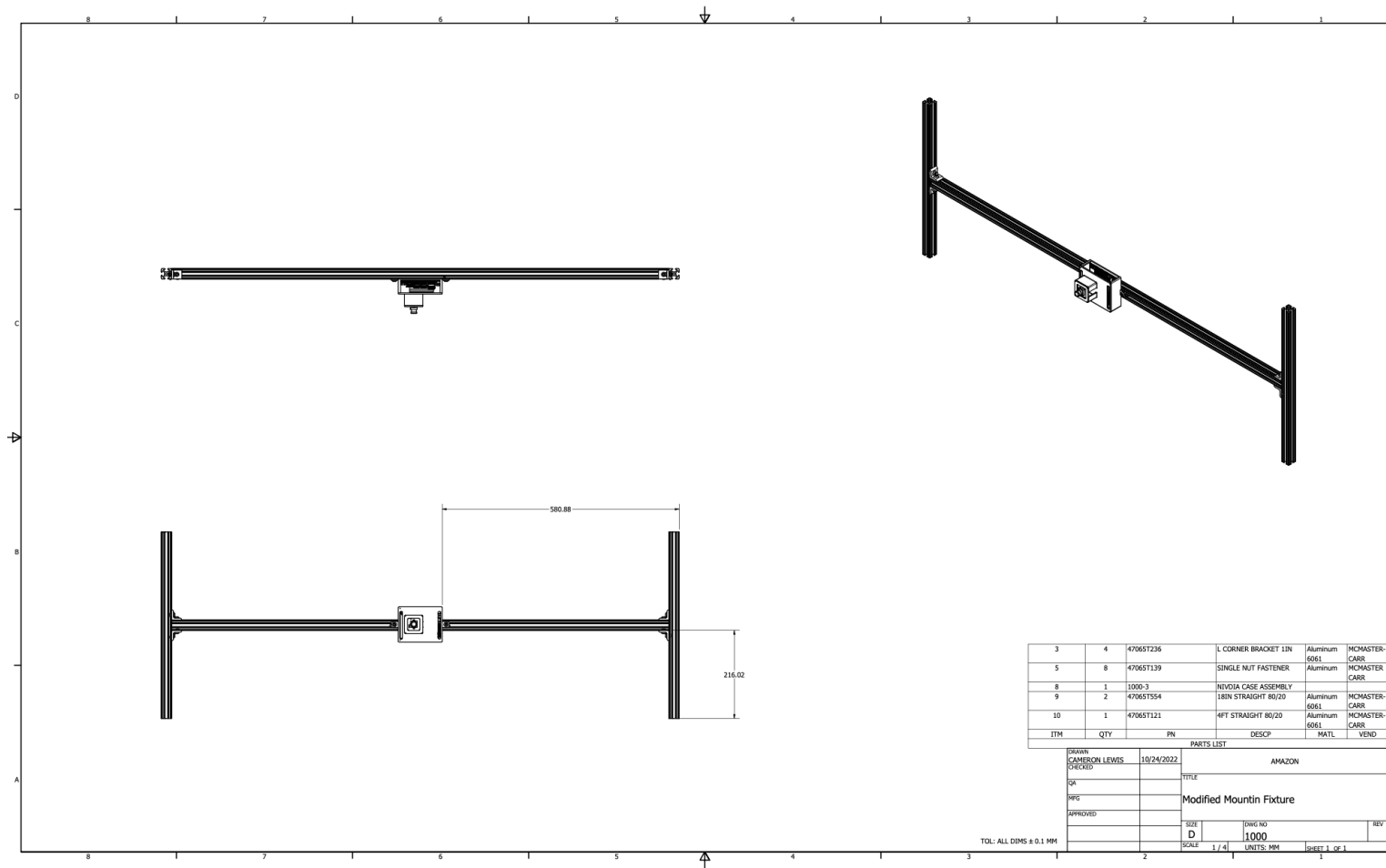
Simply change the value of the port parameter to the desired port number.

3.3. Licensing information

The Conveyor Belt Monitoring System is released under the Apache License 2.0, which is an open-source license approved by Amazon. You can find the full text of the license in the LICENSE file included in the repository. By using this software, you agree to abide by the terms of the Apache License 2.0.

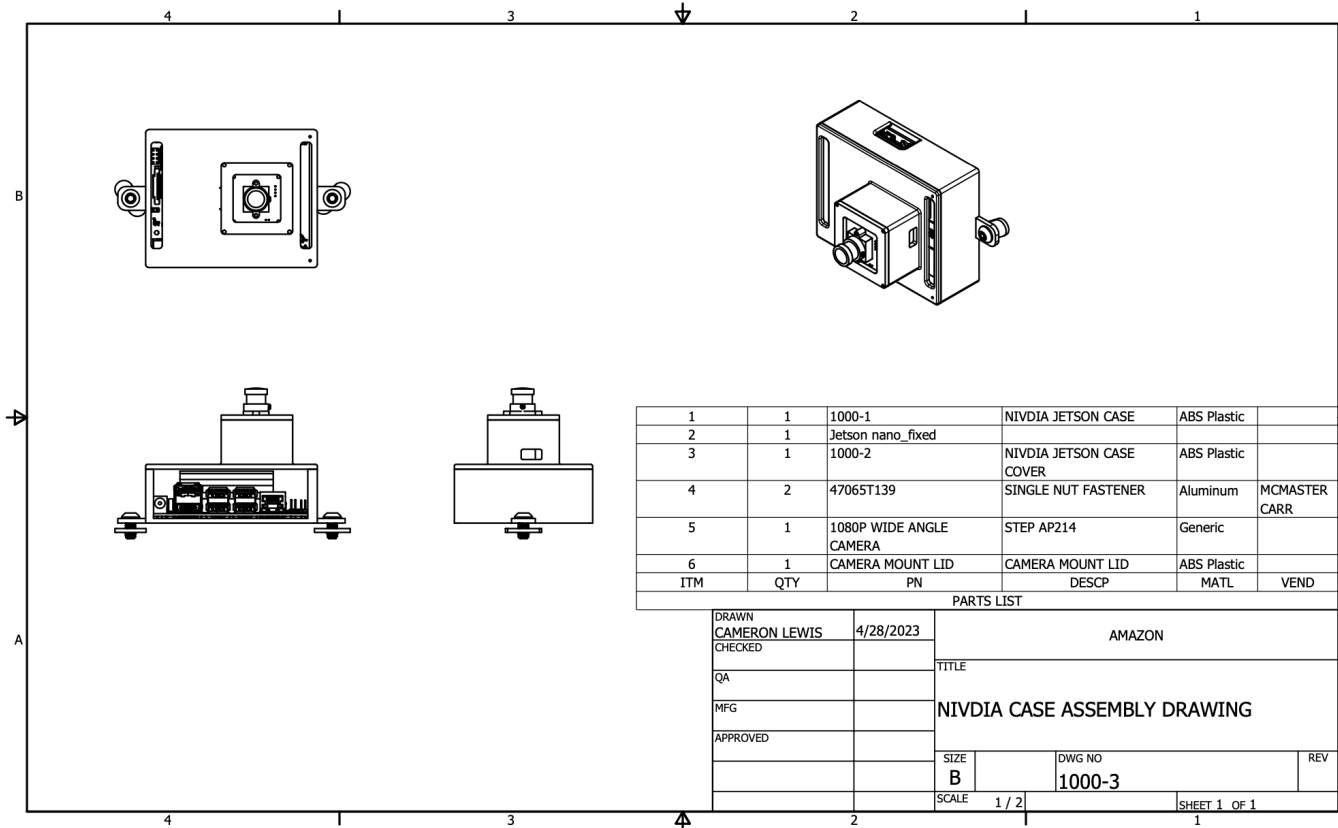
Fixture and Hardware Mechanical Designs

These downloadable CAD files and PDF of these drawings can be found at this link,

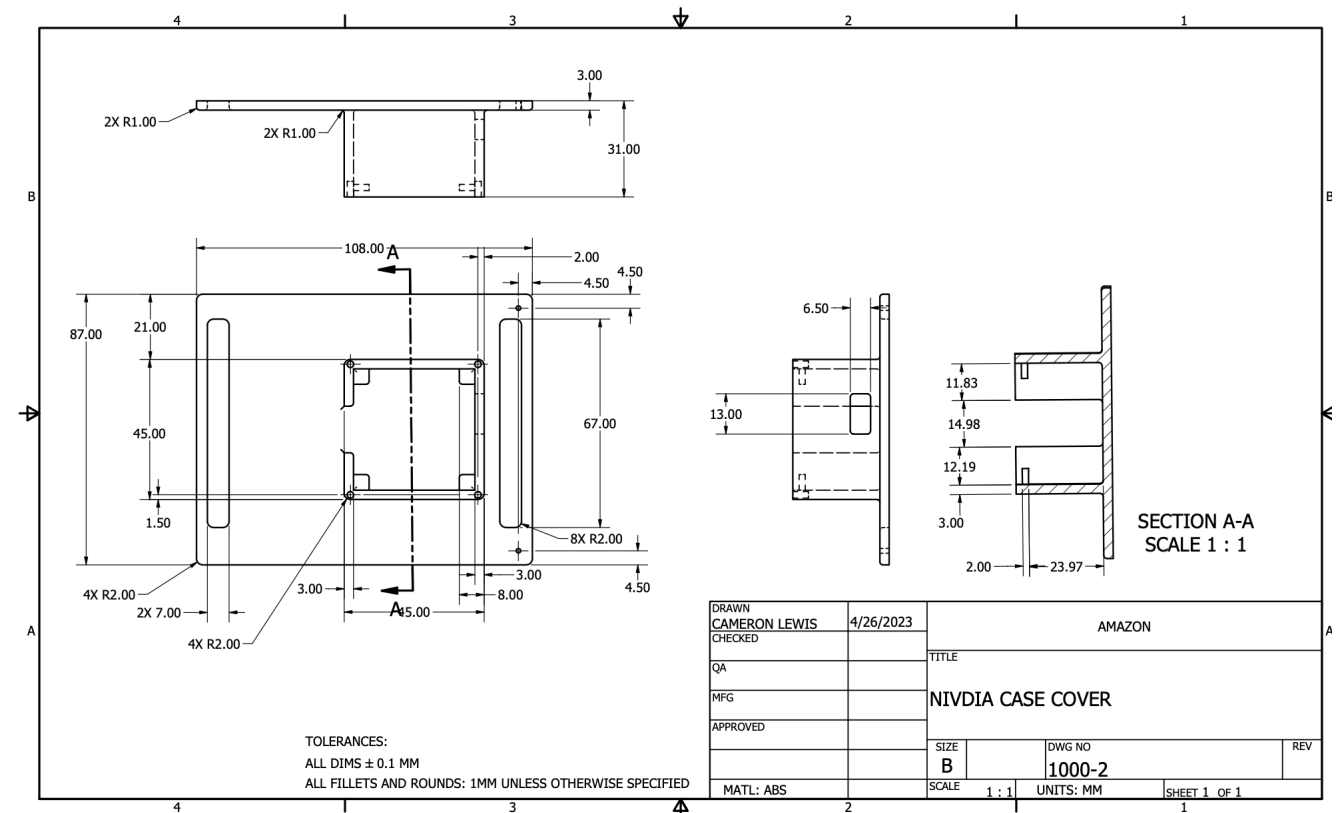


4.1. Full Assembly - Fixture & Nvidia Jetson Case

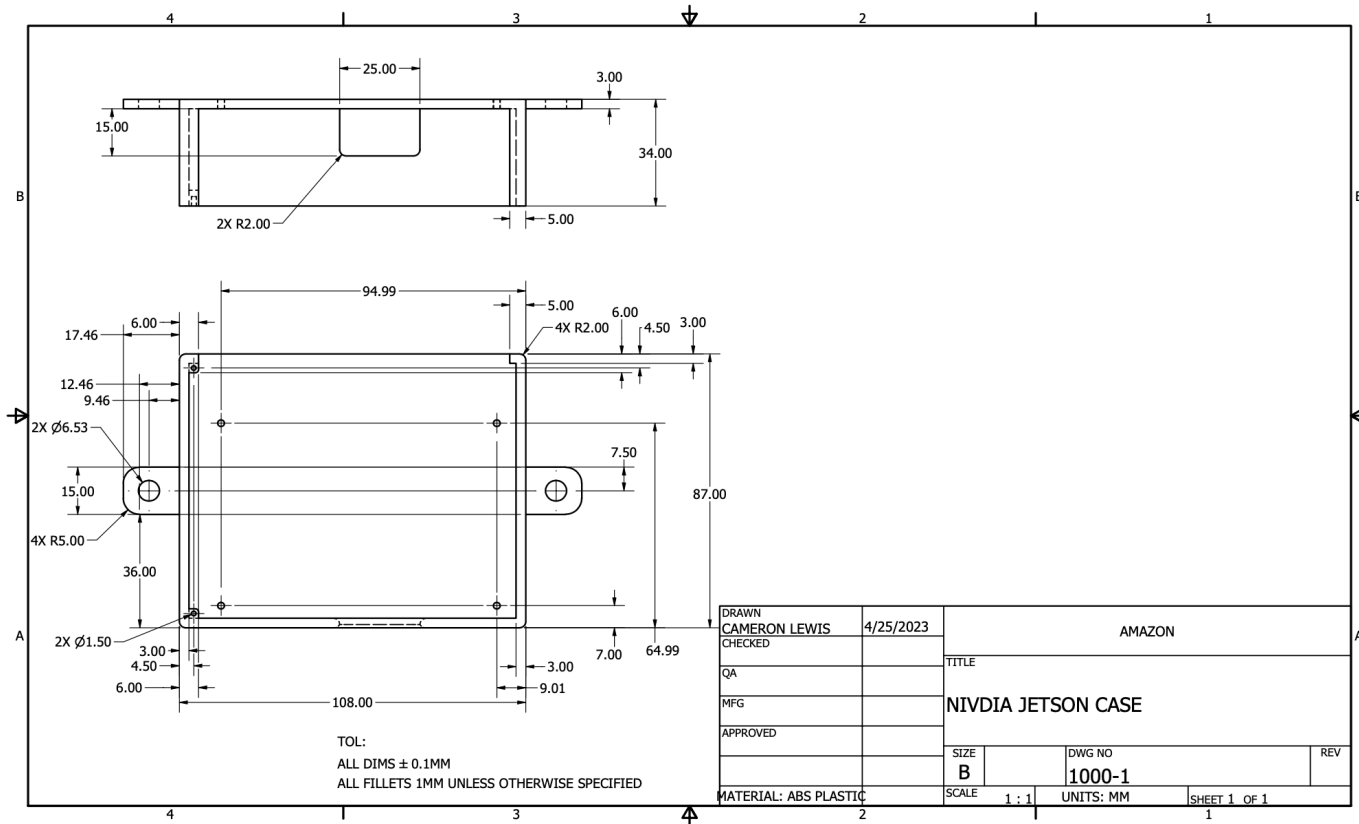
4.2. Nvidia Jetson Case - Assembly



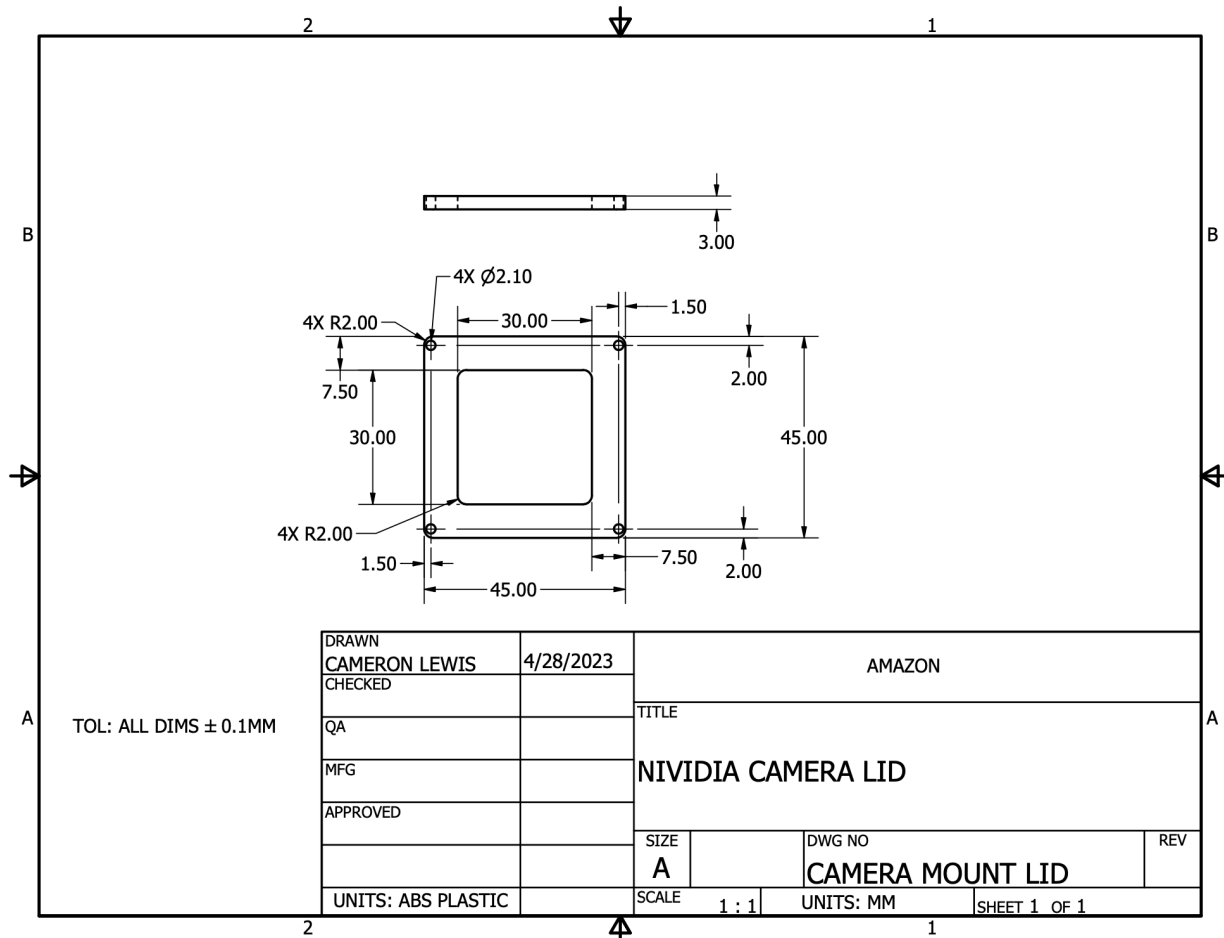
4.3. Nvidia Jetson Case - Jetson



4.4. Nvidia Jetson Case - Camera



4.5. Nvidia Jetson case - Mount Lid



User Interface

5.1. Overview of the UI

The UI (user interface) for the automated maintenance inspection system for Amazon conveyor belts is designed to be intuitive and user-friendly. It includes forms for entering data and updating benchmark values, as well as buttons for navigating to different sections of the system. The UI also includes reports and data visualizations for displaying the results of inspections.

5.2. Navigation and menus

The UI features a navigation bar with links to different sections of the system, such as the home page, the data page, and the results page. The forms for entering data and updating benchmark values are located on the home page. The results page displays the results of inspections in a table format.

5.3. Dashboard and widgets

The UI does not include a traditional dashboard, but it does feature a table on the results page that displays the performance metrics for the conveyor belt. The table includes values for average straightness, total number of rips, and average overall surface condition, as well as benchmarks for each metric. The table also includes a column for performance, which indicates whether the conveyor belt is in good or bad condition based on the benchmarks.

5.4. Forms and input controls

The UI includes two forms for entering data and updating benchmark values. The data entry form includes fields for entering the belt speed and length, while the benchmark update form includes fields for updating the benchmarks for average straightness, total number of rips, and average overall surface condition. Both forms include input controls in the form of number input fields.

5.5. Reports and data visualization

The UI includes a report in the form of a table on the results page that displays the performance metrics for the conveyor belt. The table is designed to be easy to read and includes a column for performance, which makes it easy to determine whether the conveyor belt is in good or bad condition. The UI does not include any data visualization beyond the table.

5.6. Index.html

The index.html page is the home page of the automated maintenance inspection system for Amazon conveyor belts. It provides an overview of the system and includes forms for entering data and updating benchmark values. The page is designed to be user-friendly, with a clear navigation bar and intuitive layout.

Automated Maintenance Inspection for Amazon Conveyor Belts

Generate a Maintenance Report in this column to see a detailed breakdown of the belts condition along with the images used in the algorithm.

Conveyor Belt Configuration

Belt speed (mph):12.0

Belt length (feet):15.0

Start

This program automatically runs everyday at 10am with the values set in the column below. They can be modified to the belts length, speed, and condition.

Update Benchmark Values

Speed (mph):12.0

Length (feet):15.0

Straightness Min:80

Straightness Max:100

Total Rips Max:10

Vertices Min:500

Vertices Max:1000

Update Benchmark Values

The top of the page includes the system title and a brief description of its purpose. Below that, there are two forms: one for entering data and one for updating benchmark values. The data entry form includes fields for entering the belt speed and length, while the benchmark update form includes fields for updating the benchmarks for average straightness, total number of rips, and average overall surface condition. Both forms include input controls in the form of number input fields. Below the forms, there is a brief explanation of how the system works and what it does. The page also includes a section on the importance of regular maintenance for conveyor belts, with some statistics and facts to back up the claims.

5.7. Results.html

The results.html page displays the results of inspections for the Amazon conveyor belts. It features a table that shows the performance metrics for the conveyor belt, including the average straightness, total number of rips, and average overall surface condition. The table also includes benchmarks for each metric and a column for performance, which indicates whether the conveyor belt is in good or bad condition based on the benchmarks.

At the top of the page, there is a navigation bar with links to different sections of the system. Below that, there is a summary of the results of the latest inspection, including the date of the inspection, the number of rips detected, and the overall condition of the conveyor belt. The table is located below the summary and is designed to be easy to read. It includes color-coded rows to indicate whether the performance metric is above or below the benchmark. The performance column uses green text for good condition and red text for bad condition, making it easy to see at a glance whether the conveyor belt needs maintenance.

Predictive Maintenance Report

The belt is in bad condition and requires maintenance.

Performance

Metric	Value	Benchmark	Performance
Average straightness value	26.029594481300947	80-100	Bad
Total number of rips	15	less than 10	Bad
Average overall surface condition	42.391304347826086 vertices	500-1000 vertices	Bad

Images used in the algorithm



5.8. Data.html

The data.html page displays the data entered into the automated maintenance inspection system for the Amazon conveyor belts. It features a table that shows the belt speed and length entered for each inspection, as well as the benchmarks for average straightness, total number of rips, and average overall surface

HomeRun a new ReportDatabase

Data									
ID	Average Straightness	Blue Values	Average Number of Vertices	Average Solidity	Belt Speed (mph)	Belt Length (feet)	Date	Time	
1	64.4937230768668	[0, 21070, 16145, 15460, 15413, 15313, 14552, 14525, 16635, 18485, 19019, 19554, 15320, 14418, 20192, 17577, 18888, 18587, 21008, 17426, 16205, 15980, 19703, 16440, 18328, 15955, 15805, 18111, 19344, 15418, 15989, 15780, 17169, 17688, 17831, 17885, 15275, 15667, 18891, 17250, 19417, 16449, 19610]	51.9	0.15113566462421807	12.0	15.0	2023-04-16	16:56:26	
2	64.4937230768668	[0, 21070, 16145, 15460, 15413, 15313, 14552, 14525, 16635, 18485, 19019, 19554, 15320, 14418, 20192, 17577, 18888, 18587, 21008, 17426, 16205, 15980, 19703, 16440, 18328, 15955, 15805, 18111, 19344, 15418, 15989, 15780, 17169, 17688, 17831, 17885, 15275, 15667, 18891, 17250, 19417, 16449, 19610]	51.9	0.15113566462421807	12.0	15.0	2023-04-16	16:56:32	
3	27.12343344571727	[1637, 4910, 7791, 9934, 9745, 8770, 7163, 8039, 12145, 9269, 9221, 8947, 9543, 9515, 9895, 9356, 14308, 9184, 12511, 10907, 10952, 12551, 9279, 9307, 15900, 14726, 16842, 9887, 8692, 9105, 8588, 13080, 12460, 12667, 13643, 13164, 8958, 15853, 15975, 7224, 7812, 5836, 9946, 11691, 8336, 8286, 11059, 10591, 10469, 10627, 10738, 10186, 11700, 10858, 11055, 11377, 11578, 11292, 11506, 12259, 11224, 11635, 12386, 11645, 11691, 12893, 10416, 10392, 11413, 11841, 11603, 11485, 11924, 10982, 12986, 7106, 7172, 9968, 10240, 10817, 9686, 9346, 10761, 8740, 8293, 11145, 11003, 9285, 9314, 13338, 11769]	51.06896551724138	0.14543818794164548	12.0	15.0	2023-04-16	16:56:50	

condition.

At the top of the page, there is a navigation bar with links to different sections of the system. Below that, there is a summary of the data entered into the system, including the number of inspections and the average belt speed and length. The table is located below the summary and is designed to be easy to read. It includes color-coded rows to indicate whether the benchmarks have been met or not. The table also includes buttons for editing or deleting data, making it easy to update the system as needed.

Data Management

6.1. Data sources and integration

The Conveyor Belt Monitoring System is designed to work with a variety of data sources and integrates seamlessly with Amazon's existing fulfillment center infrastructure. The system captures images of the

conveyor belt using a camera mounted above the belt and processes these images using OpenCV functions. The system then extracts key features of the belt, such as its straightness, the number of rips, and the number of vertices on its surface. The system also allows for manual data entry of the belt speed and length.

6.2. Data import and export

The Conveyor Belt Monitoring System includes data import and export features that allow users to import and export data from the system. The system allows users to export inspection reports and historical data in the datapage, accessible via SQLite3. The system also allows users to import data into the system, including belt speed and length, and benchmark values.

6.3. Data validation and cleansing

The Conveyor Belt Monitoring System includes data validation and cleansing features that ensure the accuracy and completeness of the data entered into the system. The system validates the data entered into the data entry form to ensure that it is within acceptable ranges and formats. The system also includes a data cleansing feature that identifies and removes any duplicate or inconsistent data.

Functionality

The Conveyor Belt Monitoring System provides various features and modules to automate the process of monitoring the condition of conveyor belts used in manufacturing or logistics facilities. These features and modules are designed to improve efficiency, reduce downtime, and prevent accidents.

7.1. Features and modules

The following features and modules are included in the Conveyor Belt Monitoring System:

- Image acquisition: The system captures images of the conveyor belt using a camera mounted above the belt.
- Image processing: The system processes the captured images using OpenCV functions to extract key features of the belt, such as its straightness, the number of rips, and the number of vertices on its surface.
- Inspection report: Based on the extracted features, the system generates an inspection report that indicates whether maintenance is required for the belt.
- Historical data: The system stores inspection reports in a SQLite database, allowing users to view the historical data of previous inspections.

7.2. Use cases and scenarios

The Conveyor Belt Monitoring System can be used in various scenarios, such as:

- Preventative maintenance: By monitoring the condition of the conveyor belts, the system can help identify issues before they cause downtime or accidents.
- Quality control: The system can help ensure that the conveyor belts are functioning properly and producing high-quality products.
- Compliance: The system can help ensure compliance with safety regulations and standards.

7.3. Business logic and rules

The Conveyor Belt Monitoring System follows certain business logic and rules to ensure accurate and reliable results. These rules include:

- Threshold values: The system uses threshold values for key features, such as straightness and number of rips, to determine whether maintenance is required for the belt.
- Benchmarks: The system compares the performance metrics of the conveyor belt, such as average straightness and total number of rips, to benchmarks to determine whether the conveyor belt is in good or bad condition.
- User roles and permissions: The system uses different user roles and permissions to ensure that only authorized users can perform certain actions, such as updating benchmark values.

7.4. Scheduled processing:

The Conveyor Belt Monitoring System includes a scheduled processing functionality to trigger the /process, /results, and /data routes at 10am. This is accomplished using the APScheduler library. The default speed and length values for processing can be set, and the process function generates an inspection report based on these values.

Note: The Conveyor Belt Monitoring System does not have any APIs or integrations at this time.

Maintenance and Troubleshooting

8.1. System Maintenance Tasks

The Conveyor Belt Monitoring System requires regular maintenance to ensure it runs smoothly and efficiently. Here are some tasks that should be performed regularly to keep the system in good working order:

- Check the camera and lighting system: Make sure the camera is clean and properly positioned to capture clear images of the conveyor belt. Check that the lighting system provides sufficient light for

clear image capture.

- Check the hardware components: Inspect the Nvidia Jetson Nano and the USB camera for any signs of wear or damage. Check that all cables are securely connected.
- Check the software components: Ensure that all software components are up-to-date and functioning correctly. This includes the operating system, Python, Flask, OpenCV, NumPy, and SQLite.
- Backup the database: Regularly backup the SQLite database that stores the inspection results and historical data to prevent data loss.
- Schedule maintenance: Schedule regular maintenance to clean and replace worn or damaged components such as the conveyor belt or camera.

8.2. Performance Tuning and Optimization

To optimize the performance of the Conveyor Belt Monitoring System, consider the following:

- Increase processing speed: Upgrade the hardware components such as the Nvidia Jetson Nano or USB camera to increase processing speed.
- Optimize software configuration: Tune the configuration of the software components such as OpenCV or Flask to optimize their performance.
- Limit background processes: Limit the number of background processes running on the system to reduce CPU usage.
- Optimize the camera position: Adjust the camera position and lighting system to optimize image capture and processing.

8.3. Error Messages and Logs

The Conveyor Belt Monitoring System logs error messages to help identify and diagnose problems. If an error occurs, it is important to check the logs to determine the cause of the error. The logs are stored in the log file located in the system's root directory. If necessary, additional logs can be added to the system to help identify and diagnose problems.

8.4. Known Issues and Limitations

The Conveyor Belt Monitoring System has the following known issues and limitations:

- Limited camera coverage: The system only monitors a limited area of the conveyor belt, and may not detect issues outside of this area.

- Limited fault detection: The system can only detect faults that are visible in the captured images, and may not detect hidden or subtle faults.
- Limited diagnostic capability: The system provides a general assessment of the condition of the conveyor belt, but may not provide detailed diagnostic information about specific faults.
- Limited customization: The system is designed to work with the specific hardware and software components outlined in the system requirements, and may not be easily customizable for other configurations.

Note: It is important to keep these limitations in mind when using the Conveyor Belt Monitoring System, and to perform additional maintenance and troubleshooting as necessary to ensure the system is functioning correctly.

Appendix

9.1. Glossary

Here are some technical terms and acronyms used throughout this manual:

- Computer Vision: A field of study that focuses on enabling machines to interpret and understand visual data from the world around them.
- OpenCV: An open-source computer vision and machine learning software library that provides a wide range of image processing functions.
- Flask: A lightweight web application framework that provides tools for building web applications quickly and easily.
- SQLite: A lightweight relational database management system that is included with Python.
- Feature: A specific characteristic or property of the conveyor belt that is extracted by the system, such as the straightness of the belt or the number of rips on the belt.
- Maintenance: The process of repairing or replacing worn or damaged parts of the conveyor belt to ensure its proper functioning.

9.2. References and Resources

Here are some useful resources for further reading on the topics covered in this manual:

- The OpenCV documentation: <https://docs.opencv.org/>
- The Flask documentation: <https://flask.palletsprojects.com/en/2.1.x/>
- The SQLite documentation: <https://www.sqlite.org/docs.html>

- The Nvidia Jetson Nano Developer Kit page:
<https://developer.nvidia.com/embedded/jetson-nano-developer-kit>

9.3. FAQ

Q: Can I use a different camera than the 1080p 60fps USB camera specified in the manual?

A: While the system has been optimized to work with the 1080p 60fps USB camera, other cameras may work as well. However, we cannot guarantee the same level of performance or accuracy with other cameras.

Q: Can I use a different operating system?

A: Yes, it can run anywhere as long as the necessary software libraries are installed (via pip)

Q: Can I modify the source code of the Conveyor Belt Monitoring System?

A: Yes, the source code is open-source and available on the GitHub repository provided in this manual. However, any modifications made to the system are at your own risk and may affect its performance or accuracy.

Q: Who do I contact for technical support or troubleshooting?

A: If you encounter any technical issues or have questions about the system, please contact the Amazon Analytics team with the code, the device, and this user manual. If anything becomes unreachable and no solution is clear, contact Brady Wyniemko (bwyniemko@gmail.com) or 248-877-4099.