Fault Detection on Auto mobile parts using Deep Neural Network

Final Year Project

Session 2020-2024

A project submitted in partial fulfillment of the degree of

BS in Artificial Intelligence

Department of Software Engineering

Faculty of Computer Science & Information Technology

The Superior University, Lahore

Spring 2021

Type (Nature of project)	[✓] **D**evelopment [ ] **R**esearch [ ] **R&D**
Area of specialization	Deep Neural Network
FYP ID	BSAI-FYP-S20-002

Project Group Members

Sr.#	Reg. #	Student Name	Email ID	*Signature
(i)	BAIM-S20-002	Mohammed Usamah	BAIM-S20-002@superior.edu.pk	

*The candidates confirm that the work submitted is their own and appropriate credit has been given where reference has been made to work of others

Plagiarism Free Certificate

This is to certify that, I Mohammed Usamah S/D Ghulam Murtaza, group leader of FYP under registration no BSAI-FYP-S20-002 at Software Engineering Department, The Superior College, Lahore. I declare that my FYP report is checked by my supervisor.

Date: 07/06/23 Name of Group Leader: Mohammed Usamah Signature: _____________

Name of Supervisor: Sir Hafiz Muhammad Shahzad Co-Supervisor:

Designation: Asst. Professor Designation:

Signature: ________________ Signature: _________________

HoD: Dr. Tahreem masood

Signature: _______________

Project Report

Fault detection on auto parts by using Deep Neural Network

Change Record

Author(s)	Version	Date	Notes	Supervisor's Signature
Mohammed Usamah	1.0		<Original Draft>	
Sir Hafiz Muhammad Shahzad	2.0		<Changes Based on Feedback from Supervisor>	
Sir Tayab	3.0		<Changes Based on Feedback From Faculty>	
Sir Naeem	4.0		<Changes Based on Feedback From Faculty>	
Mohammed Usamah	4.1		<Added Project Plan>	
Mohammed Usamah	4.2		<Chapter added>	
Sir Hafiz Muhammad Shahzad	4.2		<Changes Based on Feedback from Supervisor>	
Sir Tayab	5.0		<Changes Based on Feedback From Faculty>	
Mohammed Usamah	5.1		<Chapter Added>	
Sir Hafiz Muhammad Shahzad	6.0		<Changes Based on Feedback From Faculty>	

APPROVAL

PROJECT SUPERVISOR

Comments: ____

____

Name:______________________________

Date:______________________________ Signature:__________________________

PROJECT MANAGER

Comments: ____

____

Date:______________________________ Signature:__________________________

HEAD OF THE DEPARTMENT

Comments: ____

____

Date:______________________________ Signature:__________________________

Dedication

This work is dedicated to my parents for their unwavering support and sacrifices, as their dedication shaped me into the person I am today. I am also grateful for the guidance and mentorship provided by my supervisor, whose unwavering belief in my abilities has propelled me towards success. Lastly, I extend my appreciation to my superior university for providing a nurturing environment that has fostered my personal and academic growth.

Acknowledgements

I am really thankful to my supervisor who has my parents for their unwavering support and guidance throughout my academic journey. Their love and encouragement have been the driving force behind my achievements. I would also like to express my heartfelt appreciation to my supervisor and the esteemed faculty at Superior University. Their expertise and mentorship have shaped me into a well-rounded individual, and I am indebted to them for their invaluable contributions.

Executive Summary

The manufacturing of high-quality car parts is a critical aspect of the automotive industry. However, manual inspection methods used in the production process are time-consuming and prone to human error, leading to increased costs and decreased customer satisfaction. To address this issue, we propose to develop an automated solution for detecting damages in car parts during the manufacturing process.

In our purposed system, the parts of the vehicle will be examined through Deep Neural Network Architecture to find irregularities and faults.

1 Contents

Chapter 1

Introduction

1 Introduction

Chapter 1: Introduction

The automotive industry relies on the efficient and accurate manufacturing of high-quality car parts to maintain customer satisfaction and remain competitive in the marketplace. However, the current inspection methods used in the production process, which rely on manual inspection, are time-consuming and prone to human error. As a result, faulty parts can be missed, leading to increased costs and decreased customer satisfaction.

The detection of damages in car parts during the manufacturing process is a critical issue that needs to be addressed. An effective solution for detecting damages in real-time could significantly improve the efficiency of the production process, reduce costs, and increase customer satisfaction.

The purpose of this project is to develop a solution that utilizes advanced technologies, such as computer vision and machine learning, to detect and classify damages in car parts during the manufacturing process. By integrating this solution into the existing production line, we aim to significantly improve the accuracy and efficiency of the inspection process and ensure that only high-quality parts are produced.

1.1 Background

Traditional fault detection methods often rely on manual inspection. These approaches are time-consuming, labor-intensive, and limited in their ability to handle complex patterns and subtle faults. As a result, there is a growing need for more efficient and accurate fault detection techniques in the automotive industry [1].

Deep neural networks (DNNs) have emerged as a powerful tool in various domains, including image recognition, natural language processing, and speech recognition. DNNs excel at learning intricate patterns and extracting meaningful features from raw data [3]. This ability makes them well-suited for fault detection tasks in auto parts [2], where identifying subtle anomalies or deviations from normal behavior is essential.

1.2 Motivations and Challenges

- Increasing complexity of automotive systems demands more robust and efficient fault detection methods.
- Traditional approaches are limited in their ability to handle complex patterns and require extensive manual effort.
- Deep neural networks offer the potential to automate the fault detection process and improve accuracy.

1.3 Goals and Objectives

The goal of this project is to develop an automated and accurate fault detection system for automobile parts during the manufacturing process. By leveraging advanced technologies such as computer vision and machine learning, the objectives are as follows:

I. Develop a robust deep neural network (DNN) model capable of accurately detecting and classifying damages in automobile parts [4].
II. Train the DNN model using images dataset of labeled car part, encompassing various types of damages and anomalies.
III. Implement the fault detection system in real-time, integrating it seamlessly into the existing production line.
IV. Enhance the efficiency of the inspection process by reducing the reliance on manual inspection, thereby saving time and reducing human error.
V. Improve the overall quality control measures by ensuring that only high-quality car parts are produced and faulty parts are detected and addressed promptly.
VI. Conduct extensive testing and evaluation of the fault detection system to validate its accuracy, reliability, and scalability.
VII. Optimize the system for performance, making it capable of handling the high-speed production line demands of the automotive industry.
VIII. Provide a user-friendly interface for operators to interact with the system and obtain real-time insights into the detected faults.
IX. Continuously monitor and fine-tune the fault detection system to adapt to new types of damages and improve its accuracy over time.
X. Ultimately, increase customer satisfaction, reduce costs associated with faulty parts, and enhance the competitiveness of the automotive industry by delivering high-quality car parts.

By achieving these objectives, the project aims to revolutionize the automobile part manufacturing process by introducing an automated and efficient fault detection system that ensures the production of reliable and defect-free car parts.

1.4 Existing Solutions / Competitive Analysis

The inspection of automobile parts during the manufacturing process is currently carried out through manual inspection [1]. This process is often time-consuming, prone to human error, and not capable of handling the high volume of parts produced.

Manual inspection by technicians involves a visual inspection of each part, during which the technician identifies and records any damages. This process is labor-intensive and can take several hours to complete, slowing down the production process and increasing costs [1] [2].

The limitations of the existing inspection methods have significant impacts on the efficiency of the production process and the quality of the parts produced. Inaccurate inspection results can lead to the production of faulty parts, which can negatively impact customer satisfaction and increase the costs associated with recalls and repairs.

The limitations of the existing inspection methods have significant impacts on the efficiency of the production process and the quality of the parts produced. Inaccurate inspection results can lead to the production of faulty parts, which can negatively impact customer satisfaction and increase the costs associated with recalls and repairs.

1.5 Proposed Solution

Our proposed solution is an AI-based system for detecting damages in car parts during the manufacturing process. The system uses advanced computer vision and machine learning techniques to analyze images of the parts in real-time, identifying any damages that may exist [3] [4]. The system is designed to be integrated into the existing production process, minimizing the need for manual intervention.

The proposed solution consists of two main components: a high-resolution camera that captures images of the parts, and an AI-based algorithm that analyzes the images and identifies damages [5]. The AI-based algorithm is trained on a large dataset of images of damaged parts, allowing it to accurately detect damages in real-time. The algorithm is designed to be scalable, allowing it to be easily adapted to new types of damages as they are discovered.

The proposed solution will provide the following benefits:

I. Improved Accuracy: The AI-based algorithm is designed to detect damages with high accuracy, reducing the number of faulty parts that are produced and improving the quality of the parts produced.
II. Increased Efficiency: The real-time analysis of the parts will reduce the time and cost associated with the inspection process, increasing the efficiency of the production process.
III. Scalability: The solution is designed to be scalable, allowing it to be easily adapted to new types of damages as they are discovered.

1.6 Project Plan

The project "Default Detection on Automobile using Deep Neural Network" aims to address the need for efficient and accurate default detection in the automotive industry. By leveraging the power of deep neural networks, the project aims to develop a robust model that can effectively detect defaults in automobile components during the manufacturing process.

The project plan begins with the crucial step of data collection, where a diverse dataset of automobile images will be gathered, including both normal and default instances. The collected data will then undergo preprocessing and augmentation techniques to standardize the dataset and improve model performance.

Data labeling will be performed using annotation tools to annotate the default regions in the images, ensuring accuracy and consistency. This annotated dataset will serve as the training data for the deep neural network model.

The modeling phase involves designing and implementing a suitable deep neural network architecture for default detection. Different architectures, such as convolutional neural networks (CNNs), will be explored and optimized using hyper parameter tuning. Transfer learning techniques will also be considered to leverage pre-trained models for better performance.

Creating a user-friendly interface is another key aspect of the project plan. UI/UX designers will collaborate to develop an intuitive interface that allows users to upload automobile images and receive real-time default detection results. Visualizations and informative feedback will enhance user understanding.

The trained model will be evaluated using performance metrics such as precision, recall, and F1 score. Extensive testing will be conducted on a separate test dataset to assess the model's accuracy and robustness. Feedback from testing will be used to iterate and fine-tune the model.

Integration and deployment of the model and UI into a unified system will be performed to enable real-time default detection. Compatibility with existing manufacturing processes and inspection systems will be ensured, and the system will undergo testing in a production-like environment.

1.6.1 Work Breakdown Structure

- First Month collect the required data, label it and perform a function.
- Second- And Third-Month training and optimization of algorithm.
- Fourth Month develop a user Interface and implementation
- Fifth Month test the application
- Six Month application Deployment

1.6.2 Gantt chart

Task Name	Month					
	FEB	MAR	APRIL	MAY	JUNE	JULY
DATA Collection						
Model Traini						
Implementation						
Testing						
Final Documentation						

1.7 Report Outline

Introduction

Background

- Discuss the reliance on efficient and accurate manufacturing of high-quality car parts in the automotive industry.

- Highlight the limitations of current inspection methods that rely on manual inspection and are prone to human error.

Motivations and Challenges

- Emphasize the need for an effective solution to detect damages in car parts during the manufacturing process.
- Explain how real-time damage detection can improve production efficiency, reduce costs, and increase customer satisfaction.

Goals and Objectives

- State the purpose of the project to develop a solution using computer vision and machine learning to detect and classify damages in car parts.
- Highlight the aim to improve the accuracy and efficiency of the inspection process and ensure the production of high-quality parts.

Background

- Discuss traditional fault detection methods in the automotive industry that rely on manual inspection, expert knowledge, or rule-based systems.
- Explain the limitations of these methods in handling complex patterns and subtle faults.
- Introduce deep neural networks (DNNs) as a powerful tool for fault detection tasks, given their ability to learn intricate patterns and extract meaningful features.

Motivations and Challenges

A. Motivation:

- Explain the increasing complexity of automotive systems that demand more robust and efficient fault detection methods.
- Highlight the limitations of traditional approaches in handling complex patterns and requiring extensive manual effort.
- Discuss how deep neural networks offer the potential to automate the fault detection process and improve accuracy.

B. Motivation:

- Reiterate the increasing complexity of automotive systems and the need for robust and efficient fault detection methods.
- Highlight the limitations of traditional approaches and their labor-intensive nature.
- Emphasize the potential of deep neural networks to automate fault detection and improve accuracy.

Goals and Objectives

- Provide a clear overview of the project's goals and objectives.
- Highlight the specific objectives, such as developing a robust DNN model, training it on labeled car part images, and implementing it in real-time.
- Explain how the project aims to enhance efficiency, improve quality control, and increase customer satisfaction.
- Mention the importance of extensive testing, optimization, user-friendly interface creation, continuous monitoring, and improvement.

Existing Solutions / Competitive Analysis

- Describe the current inspection methods used in the automotive industry, such as manual inspection by technicians.
- Discuss the limitations of manual inspection, including time consumption, human error, and handling high production volumes.
- Explain the impact of inaccurate inspection results on costs, customer satisfaction, and recalls/repairs.

Proposed Solution

- Introduce the proposed solution of an AI-based system for detecting damages in car parts during the manufacturing process.
- Explain how the system utilizes computer vision and machine learning techniques for real-time analysis and damage identification.
- Highlight the integration of the solution into the existing production process to minimize manual intervention.
- Describe the components of the solution, including a high-resolution camera and an AI-based algorithm for damage analysis.
- Discuss the scalability of the algorithm to adapt to new types of damages.

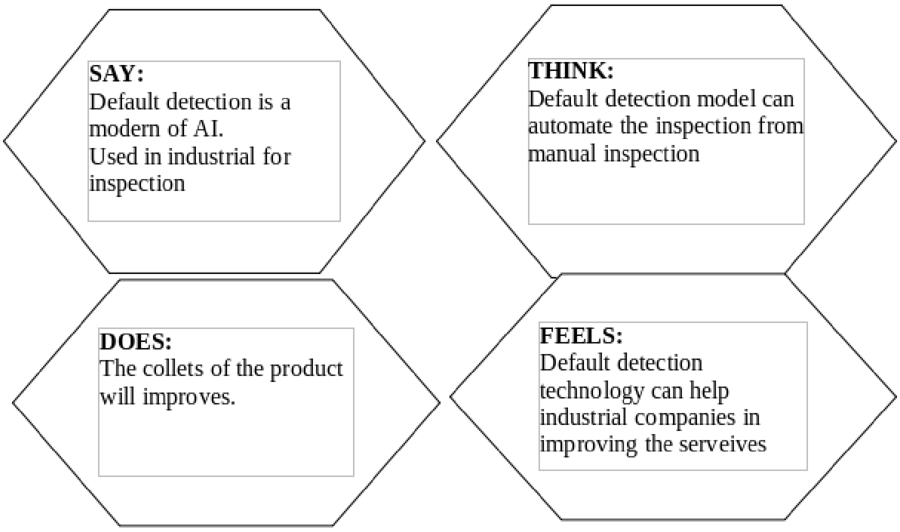
Project Plan

- Provide an overview of the project plan for developing the default detection system.
- Outline the key phases, including data collection, preprocessing and augmentation, data labeling, modeling, and creating a user interface.
- Explain the importance of each phase and how they contribute to achieving the project's goals and objectives.
- Highlight the iterative nature of the project plan, allowing for testing, evaluation, and continuous improvement.
- Mention the expected timeline for each phase and the overall project timeline.

Work Breakdown Structure

- Provide a breakdown of the project tasks and their allocation over time.
- Specify the major tasks for each month, such as data collection, training and optimization, UI development, testing, and deployment.
- Explain the logical sequence of tasks and their dependencies.
- Emphasize the need for collaboration between team members and regular progress monitoring.

1.8 Empathy Map



Chapter 2

Data Collection

2 Data Collection

2.1 Introduction

The data collection process for parts classification and default detection involved utilizing personal cameras to capture images of auto parts. Default detection was implemented using a default detection method, enabling identification of specific components. To ensure a diverse dataset, images were taken with different backgrounds, and the data was resized to 500x500 pixels for consistency. The entire data collection process was carried out in-house, emphasizing data privacy and security.

2.1.1 Sources of Data

In the process of data collection for auto parts, default detection was employed to identify specific components. To ensure a diverse dataset, personal cameras were used to capture images with varying backgrounds. This approach allowed for a comprehensive representation of different lighting conditions. Notably, the entire data collection process was carried out in-house [7] and was not outsourced to any external parties. This allowed for greater control over data quality, security, and confidentiality

2.1.2 Access the Data

To access the collected data on auto parts, a folder-based storage system was implemented. The data collected using a personal camera was organized into specific folders based on part classification and default identity [8]. This structure allowed for easy navigation and retrieval of the desired data. As the data collection process was conducted by a single person, access to the folders was solely managed by that individual. The folder-based storage system facilitated efficient analysis and research, providing a convenient way to locate and extract relevant information for auto parts development.

2.2 Data preparation

2.2.1 Part Classification

Data preprocessing plays a crucial role in the accuracy and efficiency of part classification systems. To prepare the data for analysis, the OpenCV library was utilized, providing a robust set of tools for image processing. Initially, the images were resized to a standardized dimension of 500x500 pixels [6]. This resizing ensured consistency across the dataset, enabling effective feature extraction algorithms. Additionally, to reduce the complexity of the images, a grayscale conversion was applied [10]. This conversion simplified the subsequent processing steps by eliminating color variations.

To augment the dataset, image rotation was performed, ranging from 30 degrees to 360 degrees in increments of 30 degrees [9]. This rotation introduced variations in orientation, enhancing the model's ability to recognize parts from different angles. The preprocessed images were then saved to a designated folder, ready for the labeling process. To increase the dataset size, and the number of images was significantly augmented. Default Detection

Similar data preprocessing steps were employed for default detection. The images collected using a personal camera underwent resizing to 500x500 pixels to ensure consistency. The grayscale conversion was also applied to simplify the subsequent analysis.

2.3 Data Storage

After the data preprocessing phase for parts classification and default detection, the preprocessed images were stored in a structured manner for further analysis. To facilitate model training, the data was divided into two subfolders: "training" and "validation." The split was done in an 80-20 ratio, with 80% of the data allocated to the training set and 20% to the validation set [11].

The training and validation subfolders were created to organize the data accordingly. The preprocessed images in the training subfolder were ready to be labeled for training the respective models, while the images in the validation subfolder were reserved for evaluating the model's performance.

To aid the labeling process, the images in both the training and validation subfolders were made available for use with labelImg [12]. LabelImg is a popular tool for object labeling, which would allow for the precise identification and marking of parts or defaults in the dataset.

2.4 Data labeling

For data labeling in the parts classification and default detection project, the labeling tool LabelImg was utilized. This tool enabled precise and efficient annotation of the preprocessed images [13]. By opening each image in LabelImg, experts could mark and label the parts or defaults present in the images. The tool provided a user-friendly interface with intuitive controls for drawing bounding boxes and assigning corresponding labels. This process ensured that the dataset was properly labeled, allowing the models to learn and recognize the targeted parts or defaults accurately [14].

2.5 Data privacy and security

Data privacy and security are critical considerations throughout the entire data collection, storage, and labeling process in parts classification and default detection projects. Several measures were implemented to protect the privacy and security of the data:

I. In-house data collection: By using personal cameras and conducting data collection internally, the risk of data exposure to third parties was minimized.
II. Access control: Access to the collected data was granted only to authorize personnel. This ensured that sensitive information was not accessible to unauthorized individuals.
III. Secure storage: The data was stored in a folder-based storage system, where appropriate access controls and permissions were set. This helped protect the data from unauthorized access or accidental exposure.
IV. Data anonymization: Personal identifiable information was stripped from the data during the preprocessing stage. This prevented any potential identification of individuals or sensitive information.
V. Labeling process: The data labeling process was carried out by experts who adhered to strict confidentiality guidelines. They were trained to handle the data with care and to maintain its privacy and security.

Chapter 3

Data Preprocessing

3 Data Preprocessing

3.1 Introduction

Data preprocessing is a crucial step in preparing the collected data for further analysis and model training. Leveraging the OpenCV library, advanced image processing techniques were applied. The images were resized to a standardized dimension of 500x500 pixels, ensuring consistency across the dataset. Furthermore, a grayscale conversion was performed, simplifying subsequent processing steps by removing color variations. To enhance dataset diversity, images were also rotated from 30 to 360 degrees, allowing the models to learn from various orientations.

3.2 Resizing Images

In the realm of data preprocessing for CNN models, resizing images is an essential step that brings numerous benefits to the overall analysis. This report aims to explore the process of resizing images using the cv2 library, the advantages it offers for CNN models, and the specific resizing approach of setting the data image size to 500x500 pixels [6].

3.2.1 Benefits for CNN Models

Resizing images plays a pivotal role in enhancing the performance of CNN models. Firstly, it establishes consistency in the input size, which is crucial for CNN architectures that demand fixed-size inputs [16]. This consistency enables seamless integration of the resized images into the model's input pipeline.

Moreover, resizing simplifies feature extraction by preserving the essential elements of the image while discarding unnecessary details. This process facilitates the CNN model in learning and recognizing meaningful patterns and features from the resized images.

3.3 Converting Images into Gray Scale

Converting image data to gray scale is a common data preprocessing step in various computer vision applications [17]. This report explores the benefits of using gray scale images and provides an overview of how the conversion is achieved using the OpenCV library.

3.3.1 Benefits of Gray Scale Conversion

Gray scale images contain a single channel of intensity values, resulting in lower-dimensional data compared to color images. This simplification can lead to faster computations and reduced memory requirements [10] [19].

By discarding color information, gray scale conversion eliminates the impact of varying color tones and lighting conditions. This enhances the robustness of subsequent image processing algorithms, as they can focus solely on the intensity values.

3.4 Rotation Images:

This report discusses the advantages of implementing rotation augmentation from 30 to 360 [9] degrees using the cv2 library in the context of dataset augmentation. The rotation technique diversifies the dataset by representing different object orientations, improving the generalization of models and their performance in real-world scenarios [18].

3.4.1 Benefits of Rotation Augmentation:

Rotation augmentation introduces variations in object orientation, resulting in a more diverse dataset [20]. This diversity helps models generalize better and perform well across a wide range of object orientations. Rotating images enhances the model's ability to recognize objects from various perspectives [21]. This improves the model's robustness to variations in object orientation, leading to more accurate predictions. Rotating images exposes the model to different variations in object appearance. This helps the model learn robust features that are invariant to rotation, leading to more accurate and reliable feature extraction.

3.5 Saving the Data:

After applying all the filters and preprocessing techniques, the processed images are saved to a designated folder for further labeling using the labelImg tool. The labelImg tool provides a user-friendly interface for annotating objects of interest in the images with bounding boxes [13]. This step is crucial for training the YOLOv5 model, which requires labeled data for object detection.

3.6 Conclusion:

Data preprocessing plays a vital role in preparing collected data for analysis and model training. By leveraging OpenCV, we were able to apply various techniques such as resizing images to a standardized dimension, converting images to grayscale, and rotating them to enhance dataset diversity. These preprocessing steps offer several benefits, including improved model performance, robustness, and feature extraction. Additionally, saving the processed data for labeling using labelImg allows for the creation of annotated datasets necessary for training models like YOLOv5. Overall, effective data preprocessing sets a solid foundation for successful computer vision tasks and contributes to the development of accurate and reliable models.

Chapter 4
Data Labeling

4 Data Labeling

4.1 Introduction

Data labeling is a crucial step in the process of training machine learning models. It involves annotating or marking specific objects or regions of interest in a dataset to provide ground truth labels for training the models. Two common types of data labeling are object labeling and image labeling, with image segmentation being a specific technique within image labeling [22].

Object labeling involves identifying and labeling specific objects within an image, such as cars, pedestrians, or traffic signs [23]. This labeling provides valuable information to train models for object detection, classification, and tracking tasks. It enables the models to learn the characteristics and boundaries of different objects, allowing for accurate identification and localization.

Image labeling, on the other hand, focuses on labeling the entire image with a specific category or attribute. For example, labeling an image as "indoor" or "outdoor," or labeling an image of a cat as "cat" [12]. This type of labeling provides broader context and categorization for images, enabling models to understand and differentiate between different image types.

Image segmentation is a more detailed form of image labeling [24]. It involves labeling each pixel within an image to define different regions or objects. This level of annotation allows models to understand the exact boundaries and spatial relationships between objects in an image. Image segmentation is particularly useful in tasks like semantic segmentation, where the goal is to assign each pixel in an image to a specific object or class.

The benefits of object labeling are manifold. Firstly, they provide labeled data that serves as ground truth for training machine learning models [25]. This labeled data helps the models learn and generalize patterns, enabling them to make accurate predictions and classifications. Additionally, object labeling allow models to identify and localize specific objects or categories, which is essential in tasks like object detection. Overall, these labeling techniques contribute to the development of more accurate and robust machine learning models, facilitating a wide range of applications across various industries [26].

4.2 Object Labeling

Object labeling is the process of identifying and marking specific objects within an image. It plays a crucial role in training machine learning models for object detection, classification, and tracking tasks [24]. By accurately labeling objects, models can learn to recognize and distinguish different objects, enabling them to make informed decisions and predictions.

During object labeling, annotators identify and outline the boundaries of each object of interest, such as cars, pedestrians, or animals. These annotations are typically represented as bounding boxes or polygons that enclose the objects. Additionally, object labeling may involve assigning class labels to each object, specifying the category to which it belongs.

Object labeling provides valuable labeled data that serves as ground truth for training models. It allows models to learn the visual characteristics, spatial relationships, and contextual information associated with different objects. This enables the models to accurately detect and localize objects in new, unseen images.

The benefits of object labeling extend to various applications. In autonomous driving, object labeling is essential for identifying and tracking vehicles, pedestrians, and traffic signs, enabling intelligent decision-making by self-driving cars. In surveillance systems, object labeling helps detect and monitor specific objects or individuals of interest. Object labeling is also crucial in medical imaging for identifying and segmenting anatomical structures or abnormalities.

Accurate object labeling requires expertise and attention to detail. Annotators must carefully delineate the boundaries of objects and assign correct class labels. Quality control measures, such as inter-annotator agreement and regular feedback, ensure consistency and accuracy in the labeling process.

By providing labeled data with precise object annotations, object labeling contributes to the development of robust and reliable machine learning models. These models can then be deployed in various real-world scenarios, enhancing object detection, recognition, and tracking capabilities across a wide range of industries.

4.3 Enhanced Accuracy and Precision through Object Labeling

- Precise Localization: Object labeling allows for accurate and precise localization of objects within an image, enabling models to make more accurate predictions and perform detailed analysis [27].

- Contextual Understanding: Object labeling provides contextual information about the objects, such as their relationships, spatial configurations, and interactions. This contextual understanding enhances the model's ability to interpret the scene and make informed decisions [28].
- Scalability to Complex Scenes: Object labeling can handle complex scenes with multiple objects by annotating each object individually. This scalability is crucial for applications that involve analyzing scenes with numerous instances.
- Improved Model Generalization: Object labeling enables the training of models that can generalize well to unseen data. By labeling diverse instances of objects, models learn to recognize objects under different variations, enhancing their robustness.
- Task-Specific Annotations: Object labeling allows for task-specific annotations, tailoring the labeling process to the requirements of the specific application. This ensures that the models are trained on relevant and specific information.
- Increased Annotation Efficiency: Object labeling tools provide efficient workflows and automation features, making the annotation process faster and more streamlined. This efficiency saves time and reduces the manual effort required for labeling [27].

4.4 Labeling

To label the images in LabelImg, start by selecting the YOLO option from the option menu. This ensures that the bounding boxes are compatible with the YOLO format.

Next, press the 'w' key to activate the RectBox tool, which allows you to create bounding boxes around the objects for labeling. Simply click and drag the mouse to define the box around the object of interest.

Once you have created a bounding box, it should appear around the object in the image, as shown in the figure. The box will enclose the object and serve as the annotation for the object's location and size. You repeat this process for each object in the image, creating a separate bounding box for each. LabelImg provides convenient navigation and editing tools to adjust and fine-tune the boxes as needed. By following these steps, you can accurately label the images using bounding boxes, which is essential for training object detection models like YOLO.

Chapter 5

Purposed Approach

5 Purposed Approach

V.1 Introduction

The YOLO model is used for real-time object detection by training it on labeled datasets and deploying it for inference [29]. It provides bounding box coordinates and class probabilities for detected objects, which can be post-processed for further refinement. Fine-tuning, model compression, and rigorous evaluation are conducted to optimize its performance.

V.2 YOLO

YOLO (You Only Look Once) is a state-of-the-art real-time object detection algorithm that has gained significant popularity in computer vision applications [30]. YOLO operates by dividing an image into a grid and predicting bounding boxes and class probabilities directly from the grid cells. This approach allows for fast and efficient object detection, making it suitable for real-time applications.

YOLOv5 is the fifth iteration of the YOLO algorithm, further improving upon its predecessor versions. YOLOv5 introduces a more streamlined architecture, optimizing model size and inference speed while maintaining high accuracy. It leverages advancements in deep learning techniques, such as feature pyramid networks, focal loss, and anchor-based object detection, to enhance its performance [31].

One of the key advantages of YOLOv5 is their ability to detect multiple objects within an image in a single forward pass [32]. This makes them highly efficient for tasks such as object detection, tracking, and localization. Additionally, YOLOv5 models can be trained on large-scale datasets, allowing them to learn complex object representations and generalize well to unseen data [33].

YOLOv5 are widely used in various domains, including autonomous driving, surveillance, robotics, and industrial automation. Their combination of accuracy, speed, and versatility makes them valuable tools for real-time object detection tasks. Researchers and practitioners continue to explore and refine the YOLO framework to push the boundaries of object detection capabilities [34].

V.3 Advantages of YOLO over Other CNN Models

YOLO (You Only Look Once) is a popular CNN model for object detection that offers several advantages over other models:

I. Efficiency and Speed: YOLO is highly efficient and fast, enabling real-time object detection by eliminating the need for complex region proposal methods or sliding windows. It processes the entire image at once, resulting in faster inference times [31].

II. Simplicity: YOLO has a straightforward architecture, making it easier to understand and implement compared to other CNN models. It uses a single neural network for both object localization and classification [33].

III. Accurate Multi-Object Detection: YOLO excels at detecting multiple objects within an image simultaneously. It considers the entire image during the detection process, capturing objects at different scales and aspect ratios effectively [32].

IV. Contextual Understanding: YOLO incorporates rich contextual information by analyzing the entire image. This holistic approach enhances the model's ability to comprehend object relationships, spatial configurations, and interactions [34].

V. Versatility: YOLO is a versatile model that can handle various object detection tasks, including localization, tracking, and instance segmentation. Its flexibility makes it applicable to diverse applications [31].

VI. Real-Time Applications: With its fast inference speed and high accuracy, YOLO is well-suited for real-time applications such as video surveillance, live video analysis, and robotics [34].

VII. Training on Large Datasets: YOLO can be trained on large-scale datasets, enabling it to learn from diverse data variations. This capability enhances the model's generalization ability and performance in complex scenarios.

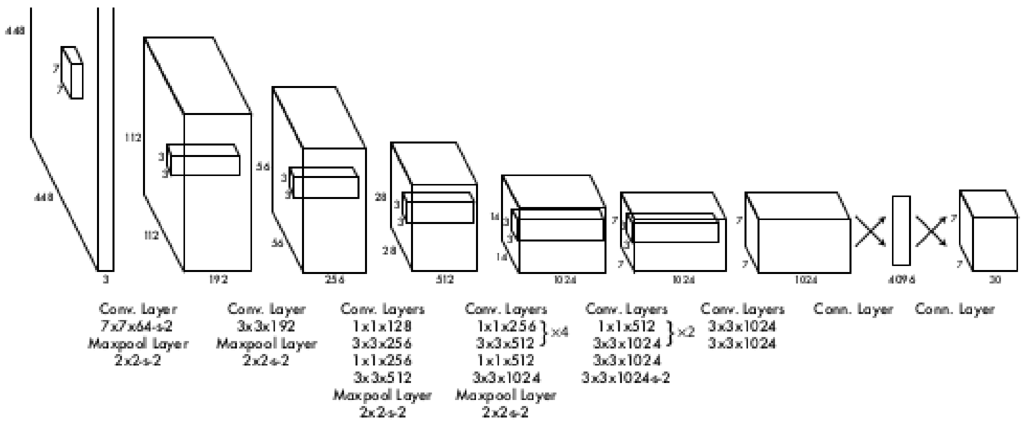
VIII. Active Development and Community Support: YOLO is actively developed and supported by a thriving community of researchers and practitioners. This ensures the model's continuous improvement and integration of the latest advancements in computer vision research.

IX. Open-Source Availability: YOLO and its variants, including YOLOv5, are available as open-source projects. This accessibility allows researchers and developers to utilize and customize the model for their specific applications [34].

X. These advantages collectively position YOLO as a powerful and efficient choice for object detection tasks, making it stand out among other CNN models in the field.

V.4 Architecture of yolo

YOLO (You Only Look Once) is a popular object detection architecture in computer vision. It revolutionized real-time object detection by introducing a single neural network that predicts bounding boxes and class probabilities directly from input images. YOLO divides the input image into a grid and assigns each grid cell the responsibility of detecting objects. The architecture utilizes convolutional layers for feature extraction and employs anchor boxes to predict multiple bounding boxes of different sizes and aspect ratios. YOLO's architecture combines speed and accuracy, making it an efficient solution for object detection tasks in various applications [30].



6 Chapter Implementation

6 Implementation

6.1 Tools and Technologies

PyTorch served as the foundational deep learning framework, offering flexibility and robust capabilities. YOLOv5, an advanced object detection model, facilitated real-time and efficient damage identification. The utilization of labeling tools played a vital role in annotating training datasets, ultimately enhancing the model's accuracy. This integration of tools and technologies facilitated a streamlined development process, resulting in a potent and precise solution for detecting damages in car parts.

6.2 Data Modeling

Partitioning the dataset into 80% for training and 20% for validation is a common practice in machine learning. This split ensures the model is trained on diverse examples and validated on distinct data to assess generalization capabilities. Applying data augmentation techniques during this process, such as rotations or flips, artificially expands the training set, enhancing the model's robustness by exposing it to a broader range of scenarios. The balanced partitioning and augmentation collectively contribute to the model's ability to effectively recognize and generalize patterns in the data.

6.3 Feature Engineering

Following data augmentation, the transformed data is converted into PyTorch tensors, as PyTorch exclusively operates with tensors. This conversion is a pivotal step in the feature engineering process, aligning the data with PyTorch's tensor-based computation framework. Tensors, the fundamental data structures in PyTorch, facilitate efficient and parallelized numerical computations. By representing the augmented data as tensors, the model seamlessly integrates with PyTorch's neural network architecture, enabling streamlined training and inference processes. This ensures compatibility and optimization within PyTorch, enhancing the overall efficiency of the feature engineering pipeline in the context of the machine learning workflow.

6.4 Model Selection and Training

YOLOv5 model is chosen for its excellence in object detection. Two YOLOv5 models are deployed: one dedicated to part detection and the other to damage detection. YOLOv5, renowned for its speed and accuracy, employs a single neural network for real-time detection tasks. Its architecture facilitates simultaneous detection of multiple objects within an image. The first YOLOv5 focuses on identifying distinct car parts, while the second is specialized in pinpointing damages within those parts. This dual-model approach optimizes precision and efficiency, ensuring a comprehensive solution for detecting both car parts and damages in the manufacturing process.

6.5 Evaluation Metrics

We leverage the Confusion Matrix to assess the performance of our YOLOv5 models. These metrics provide comprehensive insights into classification accuracy and model precision. The algorithm for computing these metrics is seamlessly integrated into the YOLOv5 training code. The Confusion Matrix aids in understanding true positives, true negatives, false positives, and false negatives. This robust evaluation strategy ensures a thorough and nuanced examination of our models' capabilities, facilitating informed decisions for further optimization and fine-tuning.

6.6 Implementation Plan

The implementation phase will leverage the powerful hardware capabilities of the ASUS TUF F15 laptop, featuring an NVIDIA GeForce 3050 Ti GPU. This GPU, with 4GB of dedicated memory, plays a vital role in accelerating the training process of our deep learning model. The development environment will be established on a Pop!_OS Linux-based system, ensuring a robust and efficient platform for AI development. Our software stack includes essential tools such as Visual Studio Code, a versatile integrated development environment (IDE), and Jupyter notebooks for interactive and collaborative coding experiences. Anaconda and PyTorch will be utilized to streamline the software setup, ensuring a smooth and productive implementation process.

7 Chapter
Results and Analysis

7 Results and Analysis

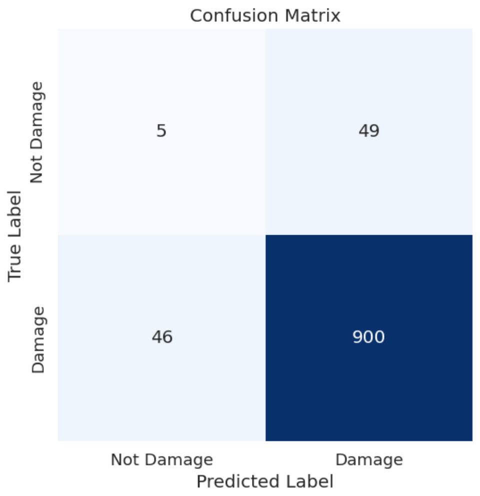
7.1 Results and Analysis

The YOLOv5 model undergoes evaluation using a Confusion Matrix. The Confusion Matrix provides insights into true positives, true negatives, false positives, and false negatives, offering a nuanced understanding of the model's performance. Live footage is incorporated for validation, ensuring the model's effectiveness in dynamically identifying car parts and damages. This multifaceted evaluation approach guarantees a robust analysis, guiding further optimizations for enhanced precision in the manufacturing process

7.2 Confusion Metrics

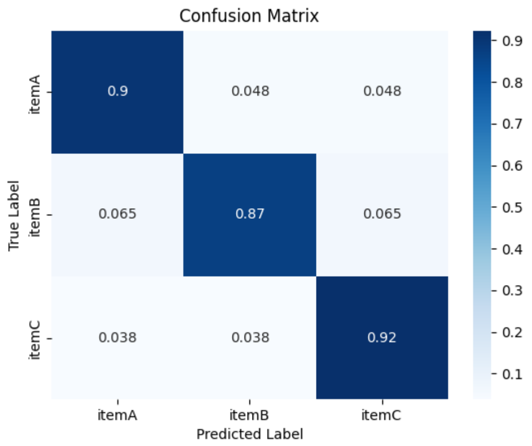
7.2.1 Damage Detection Model

The model demonstrates a remarkable 95% precision in correctly identifying instances of damages as shown in the figure



7.2.2 Parts Detection Model

The model demonstrates a remarkable in correctly identifying the Parts as shown in the figure



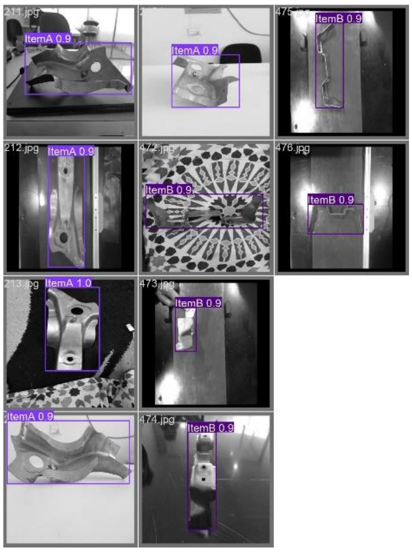
7.3 Validation Live Footage During Training

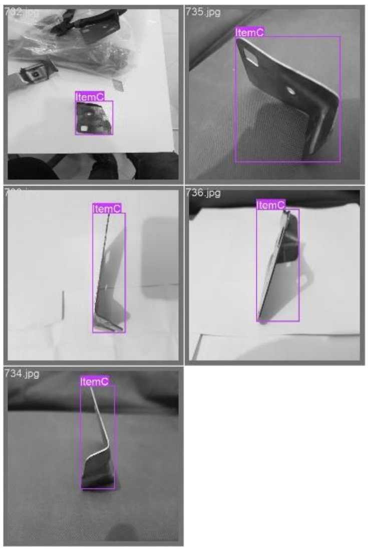
7.3.1 Damage Detection Model

During the training phase, our model not only achieved impressive statistical performance but also provided visual insights into its capabilities. The model successfully detected damaged areas on various parts. These images demonstration of the model's ability to identify and highlight specific regions indicative of damage.

7.3.2 Part Detection Model

As in Damage Detection model the Part Detection model also have identify the parts correctly.





This visual validation adds a layer of transparency to the training process, providing a clear and interpretable understanding of how the model interprets.

7.4 Ethical Considerations

The dataset used for training is obtained ethically, ensuring respect for privacy and compliance with data collection policies. The YOLOv5 model undergoes responsible training practices. Evaluation involves transparency through the application of Confusion Matrix and live training visual images, ensuring thorough assessment. As the model impacts the automobile industry, ethical implications are carefully considered, emphasizing accountability and fair usage. The prospect of reduced human errors underscores the positive ethical impact, promising advancements in automotive manufacturing with increased efficiency and accuracy, aligning with ethical principles and responsible AI deployment.

Reference

[1] Ayoub, Naeem, and Peter Schneider-Kamp. "Real-time on-board deep learning fault detection for autonomous UAV inspections." Electronics 10, no. 9 (2021): 1091.

[2] Chen, Kaijing, Wendi Wang, Xiaohui Chen, and Huarui Yin. "Deep learning based antenna array fault detection." In 2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring), pp. 1-5. IEEE, 2019.

[3] Stavrothanasopoulos, K., Gkountakos, K., Ioannidis, K., Tsikrika, T., Vrochidis, S. and Kompatsiaris, I., 2022, December. Vehicle Color Identification Framework using Pixel-level Color Estimation from Segmentation Masks of Car Parts. In 2022 IEEE 5th International Conference on Image Processing Applications and Systems (IPAS) (pp. 1-7). IEEE.

[4] Gopalakrishnan, Kasthurirangan, et al. "Crack damage detection in unmanned aerial vehicle images of civil infrastructure using pre-trained deep learning model." Int. J. Traffic Transp. Eng 8.1 (2018): 1-14.

[5] García, G., Gallardo, J., Mauricio, A., López, J. and Del Carpio, C., 2017. Detection of diabetic retinopathy based on a convolutional neural network using retinal fundus images. In Artificial Neural Networks and Machine Learning–ICANN 2017: 26th International Conference on Artificial Neural Networks, Alghero, Italy, September 11-14, 2017, Proceedings, Part II 26 (pp. 635-642). Springer International Publishing

[6] Elliott, Gavin, Kevin Meehan, and Jennifer Hyndman. "Using CNN and Tensorflow to recognise ‘Signal for Help’Hand Gestures." 2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON). IEEE, 2021.

[7] Ayoub, Naeem, and Peter Schneider-Kamp. "Real-time on-board deep learning fault detection for autonomous UAV inspections." Electronics 10, no. 9 (2021): 1091.

[8] Lukas, Jan, Jessica Fridrich, and Miroslav Goljan. "Digital camera identification from sensor pattern noise." IEEE Transactions on Information Forensics and Security 1.2 (2006): 205-214.

[9] Liu, B., Wang, X., Dixit, M., Kwitt, R., & Vasconcelos, N. (2018). Feature space transfer for data augmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 9090-9098).

[10] Dikbas, S., Arici, T. and Altunbasak, Y., 2007, September. Chrominance edge preserving grayscale transformation with approximate first principal component for color edge detection. In 2007 IEEE International Conference on Image Processing (Vol. 2, pp. II-261). IEEE.

[11] Vrigazova, B., 2021. The proportion for splitting data into training and test set for the bootstrap in classification problems. Business Systems Research: International Journal of the Society for Advancing Innovation and Research in Economy, 12(1), pp.228-242.

[12] Xiang, Y., Wang, H., Su, T., Li, R., Brach, C., Mao, S.S. and Geimer, M., 2020. Kit moma: A mobile machines dataset. arXiv preprint arXiv:2007.04198.

[13] Su, B., Zhou, Z. and Chen, H., 2022. PVEL-AD: A large-scale open-world dataset for photovoltaic cell anomaly detection. IEEE Transactions on Industrial Informatics, 19(1), pp.404-413

[14] Guo, Y., Zhang, L., Hu, Y., He, X. and Gao, J., 2016. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14 (pp. 87-102). Springer International Publishing.

[15]Zhou, W. and Peng, Y., 2023. Ship detection based on multi-scale weighted fusion. Displays, 78, p.102448.

[16] Dogra, R., Rani, S., Singh, A., Albahar, M.A., Barrera, A.E. and Alkhayyat, A., 2023. Deep learning model for detection of brown spot rice leaf disease with smart agriculture. Computers and Electrical Engineering, 109, p.108659.

[17] Gunasekaran, S., 1996. Computer vision technology for food quality assurance. Trends in Food Science & Technology, 7(8), pp.245-256.

[18] Wu, L., Shivakumara, P., Lu, T. and Tan, C.L., 2015. A new technique for multi-oriented scene text line detection and tracking in video. IEEE Transactions on multimedia, 17(8), pp.1137-1152.

[19] Bui, H.M., Lech, M., Cheng, E., Neville, K. and Burnett, I.S., 2016, July. Using grayscale images for object recognition with convolutional-recursive neural network. In 2016 IEEE Sixth International Conference on Communications and Electronics (ICCE) (pp. 321-325). IEEE

[20]Okafor, E., Schomaker, L. and Wiering, M.A., 2018. An analysis of rotation matrix and colour constancy data augmentation in classifying images of animals. Journal of Information and Telecommunication, 2(4), pp.465-491.

[21]Lowe, David G. "Object recognition from local scale-invariant features." Proceedings of the seventh IEEE international conference on computer vision. Vol. 2. Ieee, 1999.

[22]Li, Guanbin, Yuan Xie, and Liang Lin. "Weakly supervised salient object detection using image labels." In Proceedings of the AAAI conference on artificial intelligence, vol. 32, no. 1. 2018

[23] Gould, S., Gao, T. and Koller, D., 2009. Region-based segmentation and object detection. Advances in neural information processing systems, 22

[24] Cremers, D., Sochen, N. and Schnörr, C., 2006. A Multiphase Dynamic Labeling Model for Variational Recognition-driven Image Segmentation. International Journal of Computer Vision, 66(1).

[25] Brostow, G.J., Fauqueur, J. and Cipolla, R., 2009. Semantic object classes in video: A high-definition ground truth database. Pattern Recognition Letters, 30(2), pp.88-97.

[26] Dargan, S., Kumar, M., Ayyagari, M.R. and Kumar, G., 2020. A survey of deep learning and its applications: a new paradigm to machine learning. Archives of Computational Methods in Engineering, 27, pp.1071-1092.

[27] Lin, Tsung-Yi, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. "Microsoft coco: Common objects in context." In Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, pp. 740-755. Springer International Publishing, 2014

[28] Scalability to Complex Scenes: Object labeling can handle complex scenes with multiple objects by annotating each object individually. This scalability is crucial for applications that involve analyzing scenes with numerous instances

[29] Chen, J.W., Lin, W.J., Cheng, H.J., Hung, C.L., Lin, C.Y. and Chen, S.P., 2021. A smartphone-based application for scale pest detection using multiple-object detection methods. Electronics, 10(4), p.372.

[30] Redmon, J., Divvala, S., Girshick, R. and Farhadi, A., 2016. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).

[31] Liu, Zihan, Kaifeng Jing, Kai Yang, ZhiJun Zhang, and Xijie Li. "Efficient dense attention fusion network with channel correlation loss for road damage detection." IET Intelligent Transport Systems (2023).

[32] SB, B.K., Guhan, S., Kishore, M. and Santhosh, R., 2023, March. Real-time Pothole Detection using YOLOv5 Algorithm: A Feasible Approach for Intelligent Transportation Systems. In 2023 Second International Conference on Electronics and Renewable Systems (ICEARS) (pp. 1678-1683). IEEE.

[33] Wu, Y., Emotion Detection of Dogs and Cats Using Classification Models and Object Detection Model.

[34] Terven, J. and Cordova-Esparza, D., 2023. A comprehensive review of YOLO: From YOLOv1 to YOLOv8 and beyond. arXiv preprint arXiv:2304.00501.