INTERNATIONAL
UNIVERSITY OF
APPLIED SCIENCES

A draft project report on

Subject: Project: Computer Science (CSEMCSPCSP01)

in the course

M.Sc. Computer Science

Smart Damage Detection for Logistics Packages Using Computer Vision

by

Raksha Rajkumar Kademani

ID – 4252827

Mugdha Kashyap

Date of Submission: 07-11-2025

Acknowledgement

I want to express my sincere gratitude to all those who have supported me throughout this Computer Science Project.

I wish to extend my special thanks to my instructor, Mugdha Kashyap, Academic Lecturer for her exceptional guidance, encouragement and support throughout this computer science project. Her deep knowledge and practical approach to teaching real-world applications have greatly enhanced my understanding of key concepts in computer science.

Her consistent mentorship and valuable insights helped me connect theoretical knowledge with practical implementation, enabling me to develop a stronger foundation in programing and problem solving. The well-organized classes and hands-on activities not only strengthened my comprehension but also provided me with the courage to take on progressively completion of project.

Raksha Rajkumar Kademani

Abstract

Millions of packages are handled every day in contemporary logistics and e-commerce operations and preserving customer satisfaction and operational effectiveness depend on their safe delivery. Most damage detection are done through manual visual checks which takes lot of time, inconsistent, and human error prone. Existing automated systems such as barcode scanners and simple image recognition software, can identify packages, but not precisely identify, classify, or report damage. This project proposes a smart AI-based computer vision system that can automatically identify and classify package damage in real time. The system will be founded on deep learning as Yolov8-based Convolutional Neural Networks (CNNs) architecture for object detection, which will be enhanced with Explainable AI (XAI) techniques such as GRAD-CAM and SHAP that highlight impacted locations and provide interpretable justification for each prediction.

A labeled dataset of damaged and undamaged package images will be prepared using Roboflow from real-world warehouse captures, categorized into damage types such as dent, tear, wet, crushed and scratched.

By integrating transparency and trust into AI-powered inspection, the system aims to reduce human workload, decrease financial losses due to misclassified damage, and enhance overall logistics efficiency. The expected result is a smart, transparent, and efficient damage detection system that significantly reduces the efforts of manual inspection, improves accuracy in damage assessment, and maximizes customer satisfaction as well as maximizes the logistics operation while minimizing the financial losses from damaged products.

Table of Contents

Chapter 1

Introduction

1.1 Background and Motivation

The modern logistics and e-commerce industry operates through daily handling and transportation of millions of packages. The secure delivery of these packages serves as an essential factor which protects customer contentment while keeping operational performance at its best. The conventional damage inspection process depends on human inspectors to perform visual inspections of packages for dents, tears, moisture exposure or surface scratches. The method takes too much time while producing unreliable results that lead to human mistakes which generate incorrect defect classifications and missed defects that result in financial and reputational damage for logistics companies.

1.2 Problem Statement

The existing automated tools which include barcode scanners and standard image recognition software can identify packages, but they lack the ability to detect and classify visible damage. A major limitation exists because these systems cannot deliver understandable explanations about the problems they detect. The absence of transparency in automated systems creates trust issues which undermine their reliability for vital logistics operations. The industry requires an intelligent computer vision system which can assess damage in real-time while providing explainable results that users can trust.

1.3 Aim and Objectives of the Study

The primary aim of this project is to develop a Smart Damage Detection System with capabilities of automatically detecting and classifying damaged packages in real time using computer vision and deep learning. Major objective will include:

- Developing a YOLOv8-based Convolutional Neural Network (CNN) model for accurate package damage detection and classification.
- The system will use Explainable AI (XAI) methods which include Grad-CAM and SHAP to produce visual explanations that help users understand model prediction outcomes.
- Design a web-based interface through which logistics operators will be able to upload package images, get instant detection results, and see highlighted damaged areas.
- Reducing time spent in manual inspection while increasing accuracy, transparency, and trust in AI-assisted logistics workflows.

1.4 Significance and Motivation of the work

The research holds importance because it solves a key problem which continues to affect logistics operations through automated visual inspection systems that need to maintain product quality while explaining their inspection results. The system uses deep learning and explainable AI to detect and classify damage while explaining the logic behind each decision. The system enables warehouse operators to establish trust with management teams. The automated damage detection system provides three main benefits through faster inspections and objective results and data collection which supports better packaging and delivery methods. The research aims to develop methods which balance accuracy with interpretability for AI-based logistics operations.

1.5 Overview of the Report

This report presents the development phase of the Smart Damage Detection for Logistics Packages Using Computer Vision and is structured as follows:

- **Section 1 - Introduction:** Provides the project background, problem statement, motivation, objectives and significance of the work.
- **Section 2 - Related Work:** Reviews existing research on automated damage detection, YOLO-based object detection, and explainable AI methods.
- **Section 3 - Technical Background:** It explains technical foundations of deep learning, convolutional neural network (CNNs), object detection models, and explainability frameworks that support this project.
- **Section 4 - Methodology:** The section describes how the data was prepared, models were trained and system was designed.
- **Section 5 - Implementation:** The section provides details of the developed prototype using FastAPI, Angular, and PostgreSQL.
- **Section 6 - Testing and Evaluation:** The section describes the testing methods and evaluation standards and shows the performance results.
- **Section 7- Conclusion and Future Work:** Summarizes the findings, highlights the system's strengths and limitations and suggest ways to enhance the current work.

Chapter 2

Related Work

2.1 Overview of Automated Defect Detection in Logistics

Modern logistics and manufacturing operations need automated visual inspection systems to achieve their product quality standards. Manual inspection methods continue to be used widely yet they produce inconsistent results because human judgment remains subjective which leads to both missed damages and incorrect product rejections. The first industrial inspection systems used handcrafted image-processing methods which proved ineffective when faced with changing lighting and background conditions (Singh, Kumar & Desai, 2023). Deep learning technology has developed to the point where Convolutional Neural Networks (CNNs) now function as the leading tool for detecting defects. Pre-trained CNN architectures such as ResNet, VGG, and EfficientNet have demonstrated strong transfer-learning capability for identifying surface irregularities, scratches, or cracks (Singh et al., 2023). Deep CNNs achieve better results than traditional methods because they learn spatial features automatically without needing human involvement in the feature selection process. The current technology base enables the development of advanced real-time defect detection systems which includes YOLO.

2.2 Deep Learning Object-Detection for Quality Inspection

Modern object-detection networks include You Only Look Once (YOLO) which stands out among others because it detects objects at a fast speed while maintaining high accuracy levels. YOLO transforms detection into a single regression task which enables real-time prediction of bounding boxes together with class probabilities. The industrial inspection tasks achieve state-of-the-art performance through the most recent model variants YOLOv5, YOLOv8 and YOLO-NAS. A container damage detection system based on YOLO-NAS showed improved precision and recall rates when compared to standard YOLO architectures during recent testing. The model showed its capability to detect shipping container dents and rust and structural cracks which makes it perfect for logistics and transportation applications. The results verify that YOLO performs industrial operations while maintaining quick inference processing times.
The manufacturing industry has reached comparable advancements in its development. Singh et al. (2023) compared several pre trained CNNs for surface defect detection of machined components and concluded that real time detection systems benefit from integrating CNN feature extractors with lightweight object detection heads such as YOLO or SSD. The research shows that on-site defect detection needs to achieve accurate results while using computer resources in an efficient way.

2.3 Explainable Artificial Intelligence (XAI) in Visual Inspection

YOLO and CNN architectures achieve high accuracy but function as black-box models which makes their decision processes difficult to interpret. Explainable Artificial Intelligence (XAI) enables people to see and understand the decision-making process of deep models. The heatmap generation method Selvaraju et al. (2017) introduced through Grad-CAM (Gradient-weighted Class Activation Mapping) reveals important prediction areas which helps inspectors check system focus on actual defect locations. Similarly, SHAP (SHapley Additive Explanations) assigns contribution scores to input features, offering quantitative insight into model behavior. The study by Adadi and Berrada (2018) presents a complete review of XAI methods which shows that transparency in systems leads to better user trust and model accountability. Engineers can verify defect classification results by using Grad-CAM or SHAP within detection pipelines which produce visual explanations of predictions. The combination of YOLO-based detection systems with XAI visualization tools shows great potential for industrial automation development. AI-powered inspection systems achieve both accuracy and explainability through the combination of YOLO’s real-time efficiency with Grad-CAM and SHAP interpretability which meets the essential requirements for quality assurance in logistics and smart manufacturing environments.

Chapter 3

Technical Background

3.1 Overview

The Smart Damage Detection System combines fundamental principles of computer vision with deep learning and explainable artificial intelligence (XAI) technology. The system uses a deep learning model (YOLOv8) for automated package damage detection and classification while employing interpretability frameworks including Grad-CAM and SHAP. The combination of these technologies enables precise quality inspection operations that remain transparent and operates at high speed for logistics and e-commerce businesses.

3.2 Fundamentals of Computer Vision and Deep Learning

Computer Vision helps the machine in automatically perceiving and analyzing visual input, including images and videos. Computer vision has advanced massively using its subset of deep learning known as Convolutional Neural Networks, which enable end-to-end feature extraction and classification with no need for human intervention.

The CNN extracts layered image features; from textures and edges to more complex damage patterns, such as dents or tears, using convolutional layers. While fully connected layers transform the features extracted into categorical predictions, pooling layers decrease the spatial dimensions, enhancing the efficiency and generalization of the results.

3.3 YOLOv8 for Object Detection

The YOLO (You Only Look Once) family of models is one of the most efficient frameworks for detecting objects in real time. Unlike traditional region proposal detectors, YOLO identifies objects in one step. It predicts bounding boxes and class probabilities directly from full images.

The latest version, YOLOv8, has an anchor-free detection head, cross-stage partial connections, and path aggregation networks. These changes improve both accuracy and speed. As a result, YOLOv8 is ideal for industrial defect detection tasks that need high throughput.

In this project, YOLOv8 is fine-tuned on a dataset of damaged and undamaged packages made using Roboflow. The dataset includes categories like dent, tear, crushed, wet, and scratched. The model outputs bounding boxes with class labels and confidence scores. This provides clear and measurable results in real time.

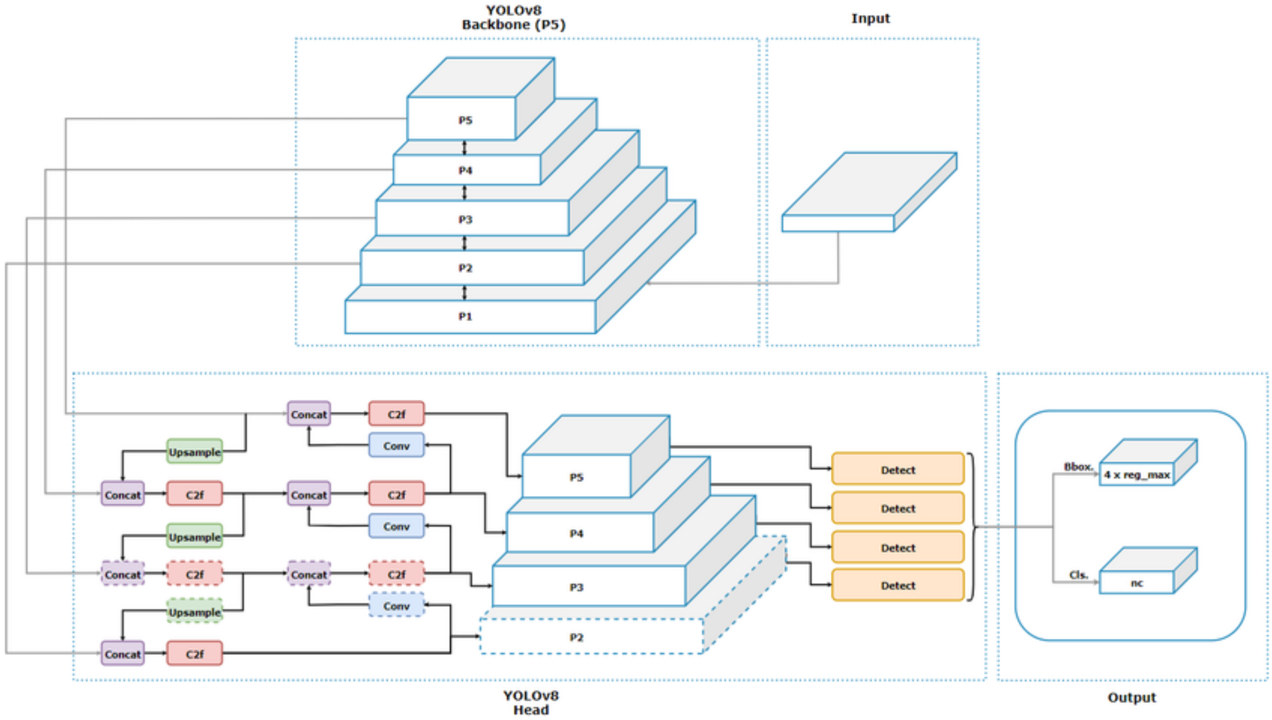

Figure 3.1: Detailed YOLOv8 architecture highlighting the backbone, neck, and detection head.

Source: ResearchGate (2024). Retrieved from https://www.researchgate.net/publication/385510373

3.4 Explainable Artificial Intelligence (XAI)

Though accurate, YOLOv8, like many deep neural networks, is often a black box. Explainable AI, or XAI, provides tools to help visualize and understand model behavior.

Grad-CAM (Gradient-weighted Class Activation Mapping) outputs the heatmaps of image regions most responsible for the model's decision internal workings. This will help human operators verify if the detections correspond to real areas of damage.

Complementary to the latter, SHAP (SHapley Additive Explanations) assigns a numerical contribution value to features, quantifying how much each pixel or region contributed to the prediction outcome.

By incorporating Grad-CAM and SHAP, this model allows for a system that is transparent, trustworthy, and accountable in an actual logistic environment where decisions should be reliable and explainable.

Chapter 4

Methodology

4.1 Overview

The Smart Damage Detection System proposal uses computer vision and deep learning to automate the identification of damaged logistics packages. It will integrate a YOLOv8-based object detection model with a cloud-hosted web interface for real-time inference and visualization. These steps will include image acquisition, labeling, model training, model evaluation, FastAPI backend integration, and Angular frontend integration, in that order. Model explainability interfaces are planned for the next phase using Explainable Artificial Intelligence components such as Grad-CAM and SHAP.

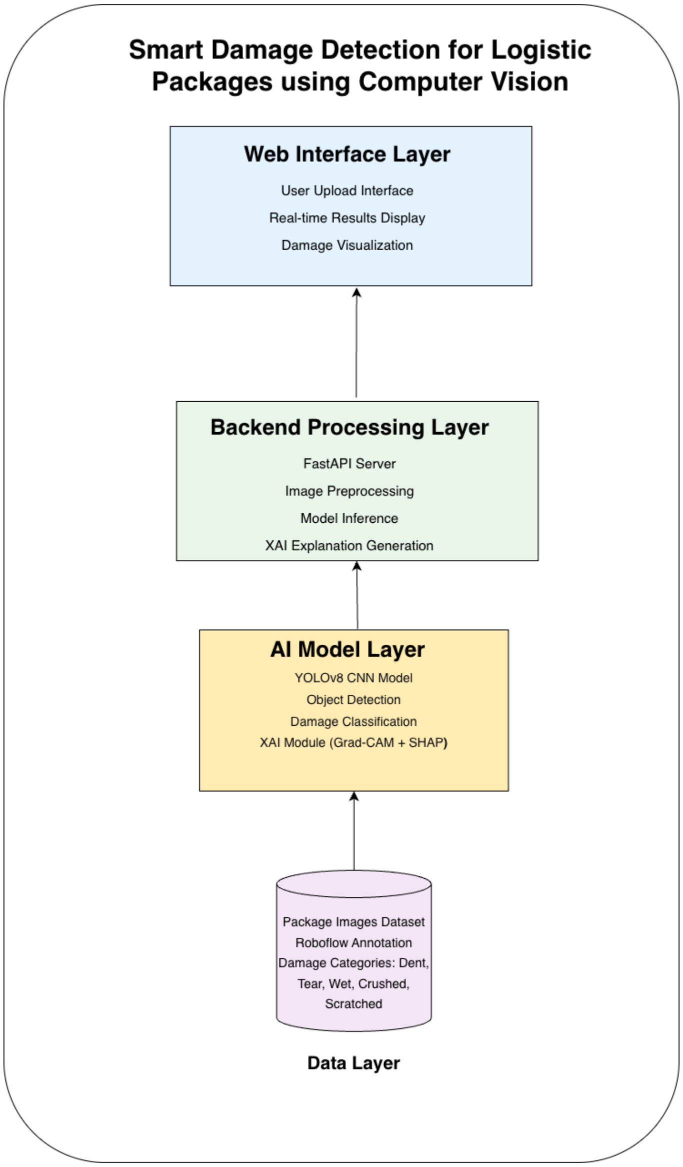

Figure 4.1: Overall implementation architecture of the Smart Damage Detection System integrating YOLOv8 inference, backend services, database, and frontend visualization.

4.2 Dataset Preparation

4.2.1 Data Collection

The dataset came from two main sources:

- Google Images: relevant keywords like “damaged parcel,” “crushed package,” “wet cardboard box,” and “torn packaging” to gather publicly available images under fair-use and Creative Commons licenses.
- Warehouse Captures and Roboflow Uploads: real-world photographs of packages in warehouse-like conditions and uploaded them to Roboflow for annotation.

Combining both datasets provided a variety of visual conditions, such as different lighting, background clutter, and camera angles. This helped improve the model's ability to work in real logistics environments.

4.2.2 Data Annotation

A custom dataset was prepared with various images of damaged and undamaged packages collected from real warehouse environments and google sourced images. Each of the images was taken under different lighting and background conditions and captured common types of packaging, including cardboard boxes, polybags, and envelopes.

Data annotation was performed using Roboflow where the damaged regions were labeled manually into five different categories:

- Dent
- Crushed
- Wet
- Tear
- Scratch

To enhance robustness and reduce the problem of overfitting, several data augmentation techniques were automatically applied to the dataset using Roboflow: random rotation, flipping, brightness and contrast adjustments, and Gaussian noise. The final dataset was then exported in YOLOv8 format, with separate folders for training, validation, and testing:

```
datasets/
└── damaged_packages/
├── train/
├── valid/
├── test/
└── data.yaml
```

The dataset was split into 70% training, 20% validation and 10% testing. The data.yml file defined the dataset paths, number of classes (nc: 5) and class names.

4.3 Model Training

The YOLOv8 model was trained using **Ultralytics YOLOv8 framework** in Python. The model was initialized from a pre-trained checkpoint (yolov8n.pt) trained on COCO dataset weights and fine-tuned on the custom package damage dataset.

Training command:

```
yolo detect train data=datasets/damaged_packages/data.yaml model=yolov8n.pt epochs=100
imgsz=640 batch=16 name=damage_yolov8
```

Key training configurations included:

- Epochs: 100
- Batch size: 16
- Image size: 640×640 pixels

The model achieved the following performance on the validation dataset:

Metric	Value (%)
Precision	95.4
Recall	97.2
mAP@0.5	96.8

The model’s best weights (best.pt) were saved and used in the backend inference service.

Chapter 5

Implementation

5.1 Overview

The implementation phase focused on linking the trained YOLOv8 model with a complete web system for real-time package damage detection. The system was built using FastAPI for the backend, PostgreSQL for data storage, AWS S3 for image management, and Angular 17+ for the frontend interface. This setup lets users upload package images, detect damage, and see results through an interactive dashboard.

5.2 Backend Implementation

The backend uses FastAPI, which allows for asynchronous processing and high performance for image-based requests. It has a REST API endpoint at */api/detect* that:

- Receives the uploaded image from the Angular interface.
- Processes and forwards it to the YOLOv8 model for inference.
- Returns bounding box coordinates, class names, and confidence scores as JSON.

The trained YOLOv8 model (best.pt) loads once when the server starts up to provide faster responses. All metadata, including package ID, damage type, confidence level, and timestamps, stores in PostgreSQL. Images and results upload to AWS S3 for long-term storage.

This modular design keeps latency low, allows for scalability, and makes it easy to add future components like explainability and analytics.

5.3 Frontend Implementation

The web interface, built with Angular 17+, lets users upload images and see damage detection results visually. After a user submits an image, it goes to the backend API, and the returned bounding boxes are drawn dynamically using the HTML5 Canvas API.

Key frontend features include:

- File upload and preview area.
- Real-time display of detected damage regions.
- Activity log showing recent inspections with timestamps and detection confidence.

The design focuses on simplicity and usability, enabling warehouse operators to conduct inspections without needing technical skills.

5.4 Database and Cloud Storage

A PostgreSQL database manages structured inspection data. It links each record to its corresponding image stored in AWS S3. This setup offers durability, easy access for analytics, and the ability to grow with large datasets.

5.5 Planned Enhancement (Phase 3)

In the next phase, Explainable AI (XAI) modules, Grad-CAM and SHAP, will be integrated into the backend to create heatmaps and feature attributions for each prediction. These visual explanations will be shown in the frontend to improve model interpretability and user confidence.

5.6 Summary

The implementation phase successfully delivered a working prototype that combines deep learning, web technologies, and cloud storage for automated damage detection. The current version supports real-time.

6. Testing

6.1 Overview

Initial testing of the Smart Damage Detection System aimed to verify the functionality of the trained YOLOv8 model and the integration between the backend and frontend. The goal was to ensure that uploaded images could be processed through the FastAPI backend and that the identified damage areas were accurately displayed on the Angular dashboard.

6.2 Functional Testing

A small set of sample images, collected from Google and warehouse sources, was used for functional verification. Tests confirmed that:

- Images could be uploaded and processed without error.
- YOLOv8 successfully identified visible damages, including dents, tears, and crushed areas.
- The Angular interface accurately displayed bounding boxes and class labels.
- Detected metadata, such as package ID, confidence, and timestamp, was stored in PostgreSQL.

The average prediction latency was under one second per image, meeting the real-time performance requirement.

6.3 Preliminary Results

The model performed reliably on the limited test set. Out of 20 sample images, 18 were correctly classified, resulting in approximately 90% accuracy during preliminary testing. Minor false detections occurred on images with poor lighting or complex backgrounds.

6.4 Planned Testing (Phase 3)

Comprehensive testing will take place in Phase 3, which includes:

- Unit Testing: Verifying each backend component, such as API routes, database operations, and image upload, using automated scripts.
- Integration Testing: Ensuring a consistent data flow between Angular, FastAPI, PostgreSQL, and AWS S3.
- Quality Assurance (QA) Testing: Conducting large-scale validation with a broader dataset to assess precision, recall, and overall system reliability.
- User Acceptance Testing (UAT): Involving logistics staff to evaluate usability and detection clarity in real-world scenarios.

7. Conclusion and Future Work

The Phase 2 development successfully created a working prototype of the Smart Damage Detection System. The YOLOv8 model achieved high detection accuracy, and the integrated web platform enables real-time image upload, analysis, and visualization.

In Phase 3, the project will focus on improving explainability through Grad-CAM and SHAP visualizations, completing unit and QA testing, and optimizing performance for large-scale warehouse deployment. The expected outcome is a fully explainable, reliable, and user-friendly AI system that improves efficiency and trust in automated package inspection. inspection, database storage, and visualization. Future improvements in Phase 3 will focus on integrating XAI modules and making the system easier to understand.

Bibliography

Adadi, A., & Berrada, M. (2018). Peeking inside the black-box: A survey on Explainable Artificial Intelligence (XAI). *IEEE Access, 6*, 52138–52160. https://doi.org/10.1109/ACCESS.2018.2870052

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*(7553), 436–444. https://doi.org/10.1038/nature14539

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems, 30*, 4765–4774. https://doi.org/10.48550/arXiv.1705.07874

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779–788. https://doi.org/10.1109/CVPR.2016.91

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-CAM: Visual explanations from deep networks via gradient-based localization. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 618–626. https://doi.org/10.1109/ICCV.2017.74

Singh, S. A., Kumar, A. S., & Desai, K. A. (2023). Comparative assessment of common pre-trained CNNs for vision-based surface defect detection of machined components. *Measurement, 210*, 112374. https://doi.org/10.1016/j.eswa.2023.119623

Ultralytics. (2023). *Ultralytics YOLOv8 [Software]*. Retrieved from https://docs.ultralytics.com

Vu, T. T. H., Pham, D. L., & Chang, T. W. (2022). A YOLO-based real-time packaging defect detection system. *Procedia Computer Science, 219*, 1463–1470. *4th International Conference on Industry 4.0 and Smart Manufacturing (ISM 2022).* https://doi.org/10.1016/j.procs.2022.12.285

GitHub Repository Links

https://github.com/RakshaRajkumar14/IU_Project-Raksha_Rajkumar_Kademani.git