

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Real-time AI-Based Computer Vision Applied to Analyse Recycled Glass Quality

Rui Jorge Ferreira Lima



Master's degree in Electrical and Computer Engineering

Supervisor: Prof. José Maria Corte Real da Costa Pereira

November 5, 2025

Resumo

Esta dissertação aborda o desafio da automatização da avaliação da qualidade do vidro reciclado, ou casco, utilizado na produção de recipientes de vidro. O trabalho é motivado pelas limitações dos procedimentos manuais de controlo de qualidade, que são morosos, sujeitos a erro humano e baseados em amostras de pequena dimensão. Esta dissertação propõe uma arquitetura deep learning para estimar a composição das amostras de casco através de análise de imagens do mesmo.

O sistema proposto recorre a uma combinação de técnicas de deteção de objetos, classificação e segmentação para estimar a composição das amostras de casco com base exclusivamente em dados visuais. Um modelo Faster R-CNN é utilizado para identificar e localizar todas as partículas visíveis. Cada região localizada é posteriormente processada por uma rede ResNet-18, que categoriza o material reciclado de acordo com nove classes previamente definidas. Para estimar a área física de cada partícula, é aplicado o algoritmo de segmentação de Chan–Vese, sendo a área em píxeis convertida em peso estimado com base em densidades específicas por classe, obtidas a partir do conjunto de treino. A distribuição do peso estimado por classe serve então de base para avaliar a qualidade da amostra, segundo os limiares de contaminação estabelecidos.

A metodologia foi desenvolvida e avaliada com base num conjunto de dados próprio, criado para este fim. Foram aplicadas técnicas de aumento de dados e calibração da câmara para melhorar a generalização e garantir consistência métrica. Os resultados experimentais demonstram que o sistema é capaz de realizar deteção e classificação de partículas com elevada precisão em condições realistas, apresentando limitações principalmente nas classes de contaminantes sub-representadas.

Abstract

This thesis addresses the challenge of automating the quality assessment of recycled glass, or cullet, used in glass container manufacturing. The work is motivated by the limitations of manual quality control procedures, which are time-consuming, subject to human error, and based on small sample sizes. A deep learning-based pipeline is proposed to estimate the material composition of cullet samples using image-based analysis.

The proposed system utilizes a combination of object detection, classification, and segmentation techniques to estimate the composition of cullet samples solely based on image data. A Faster R-CNN model is employed to identify and localize all visible particles. Each localized region is subsequently processed by a ResNet-18 network, which categorizes the recycled material according to nine predefined classes. To estimate the physical area of each particle, the Chan–Vese segmentation algorithm is applied, and the resulting pixel area is converted into an estimated weight using class-specific density values derived from the training dataset. This information enables the computation of a class-wise weight distribution, which serves as the basis for assessing sample quality in accordance with established contamination thresholds.

The methodology was developed and evaluated on a custom dataset of cullet samples created for this task. Data augmentation techniques and camera calibration were applied to enhance generalization and ensure metric consistency. Experimental results demonstrate that the pipeline can perform accurate particle detection and classification under realistic conditions, with limitations primarily affecting under-represented contaminant classes.

UN Sustainable Development Goals

The United Nations Sustainable Development Goals (SDGs) present a strategic framework to foster sustainable innovation, industrial efficiency, and environmental responsibility. This thesis, which explores the application of AI-driven computer vision to assess the quality of recycled glass in real time, aligns directly with several SDGs by addressing industrial automation, sustainable production practices, and carbon footprint reduction.

The specific Sustainable Development Goals that are advanced through this project are:

SDG 9 Build resilient infrastructure, promote inclusive and sustainable industrialization and foster innovation

SDG 12 Ensure sustainable consumption and production patterns

SDG 13 Take urgent action to combat climate change and its impacts

SDG	Target	Contribution	Performance Indicators and Metrics
9	9.4	Development of AI-based monitoring systems for quality control enhances the technological capability and innovation of the industrial sector.	Deployment of automated vision systems; Reduction in manual inspection time
	9.5	Supports applied research in AI and computer vision for industrial sustainability.	Model performance metrics (precision, recall)
12	12.5	Increases the efficiency of glass recycling processes by reducing contaminant levels and improving cullet purity.	Contaminant detection rate; Rejection rate reduction; Cullet purity percentage
	12.a	Enhances scientific and technological capacity for sustainable production.	Technical documentation and scalability assessment for industrial integration
13	13.2	Contributes to climate targets by promoting increased cullet use, thereby lowering CO ₂ emissions in glass manufacturing.	Cullet usage rate; Estimated CO ₂ reduction per tonne of cullet used

Acknowledgements

Quero agradecer a todos os meus amigos e família pelo apoio prestado durante este periodo.

Um obrigado aos meus orientadores, Dr. Paulo Jorge da Cunha Gomes e Prof. José Maria Corte Real da Costa Pereira, pela oportunidade de trabalhar neste projeto e crescer como aluno e profissional.

Rui Jorge Ferreira Lima

“Our greatest glory is not in never falling, but in rising every time we fall”

Confucius

Contents

1	Introduction	1
1.1	Context	1
1.2	Motivation	2
1.3	Objectives	2
1.4	Document Structure	3
2	Background - Scientific Principles and Manufacturing of Glass Containers	4
2.1	Glass Fundamentals - Definition and Properties of Glass	4
2.2	Glass Containers Manufacturing	6
2.2.1	Glass Container Morphology	7
2.2.2	Glass Containers Manufacturing Process	10
2.2.3	Integration of Recycled Glass in Container Manufacturing	12
2.2.4	Glass Containers Defects	18
3	Literature Review	23
3.1	2D Object Detection Methodologies	23
3.1.1	Traditional Detectors	24
3.1.2	CNN-based Architectures	25
3.2	2D Image Segmentation Methodologies	28
3.2.1	Bottom-Up Segmentation	28
3.2.2	Top-Down Segmentation	32
4	Deep Learning Approach for Automatic Cullet Quality Estimation	35
4.1	Dataset Creation	35
4.2	Automatic Cullet Quality Estimation	40
4.2.1	Multi-Object Bounding Box Proposal	41
4.2.2	Single-Object Classification	45
4.2.3	Sample Quality Estimation	47
4.2.4	Sample Weight Distribution Estimation	52
5	Experimental Results and Performance Evaluation	54
5.1	Evaluation Metrics for Object Detection	54
5.2	Evaluation Results for the Faster R-CNN Model	56
5.3	Evaluation Metrics for Object Classification	58
5.3.1	Classification Performance Evaluation	59
5.3.2	Chan-Vese Model Performance Evaluation	61

6	Conclusions	63
6.1	Future Work	64
6.2	Final Remarks	64
	References	65
A	Cullet Treatment Process Diagram	69

List of Figures

2.1	Schematic representation of the volume–temperature relationship for a liquid, supercooled liquid, glass, and crystalline solid.	5
2.2	Two-dimensional schematic of (a) a crystalline structure with long-range atomic order and (b) the disordered glassy form of the same compound. Adapted from [7].	6
2.3	Schematic Glass Container Composition. Adapted from [11].	8
2.4	Illustration of common glass container types.	9
2.5	Schematic of the glass container production line. Adapted from [12].	10
2.6	Schematic of the Blow-and-blow process. Adapted from [11].	11
2.7	Schematic of the Press-and-blow process. Adapted from [11].	11
2.8	Examples of cullet types sorted during the treatment process.	13
2.9	Schematic of the process to treat recycled glass	13
2.10	Representative examples of components identified in a flint cullet sample.	17
3.1	Comparison between image classification and object detection.	24
3.2	Architecture of a Convolutional Neural Network (CNN).	26
4.1	Experimental setup used for image acquisition.	36
4.2	Overview of the proposed image-based cullet quality estimation pipeline. The system integrates object detection, classification, and segmentation modules to compute the class-wise weight distribution for each sample image.	41
4.3	Comparison of images before and after CLAHE.	42
5.1	Visual examples of the Chan-Vese segmentation algorithm applied to different cullet particles.	62
A.1	Detailed flowchart of the cullet treatment process. The vertical layout illustrates all unit operations and material pathways used in transforming post-consumer recycled glass into furnace-ready cullet.	70

List of Tables

2.1	Flint Cullet Component Analysis	16
2.2	Representative examples of thermal stress defects in glass containers, including their causes, corrective strategies, and visual illustrations. Adapted from [11]. . .	18
2.3	Representative examples of mechanical forming defects in glass containers, including their causes, corrective strategies, and visual illustrations. Adapted from [11].	19
2.4	Representative examples of surface defects from contamination or wear in glass containers, including their causes, corrective strategies, and visual illustrations. Adapted from [11].	20
2.5	Representative examples of structural or dimensional defects in glass containers, including their causes, corrective strategies, and visual illustrations. Adapted from [11].	21
2.6	Representative examples of material or composition defects in glass containers, including their causes, corrective strategies, and visual illustrations. Adapted from [11].	22
4.1	Mapping of original to rectified class IDs.	38
4.2	Summary of training parameters and configurations.	44
4.3	Class label mapping used in the single-object classification task.	46
4.4	Summary of training parameters and configurations for the multi-class classification model.	47
5.1	Interpretation of Detection Outcomes in Terms of the Confusion Matrix	55
5.2	Faster R-CNN Performance Metrics on the Test Set	56
5.3	Confusion Matrix for Test Set Predictions (Cullet vs. Background)	56
5.4	Comparison of Validation and Test Set Performance	57
5.5	Classification Report on the Test Set	59
5.6	Confusion Matrix for ResNet-18 Classifier on Test Set	60
5.7	Classification Report on the Validation Set	60
5.8	Confusion Matrix for ResNet-18 Classifier on Validation Set	61

List of Acronyms

AAM	Active Appearance Models
AI	Artificial Intelligence
AP	Average Precision
ASM	Active Shape Models
BA	BA Glass (Company name)
BCE	Before the Common Era
CE	Common Era
CLAHE	Contrast Limited Adaptive Histogram Equalization
CNN	Convolutional Neural Network
COCO	Common Objects in Context (dataset)
CVPR	Conference on Computer Vision and Pattern Recognition
ECCV	European Conference on Computer Vision
FC	Fully Connected (Layer)
FEUP	Faculdade de Engenharia da Universidade do Porto
FIFO	First-In, First-Out
FN	False Negative
FP	False Positive
FPN	Feature Pyramid Network
HOG	Histogram of Oriented Gradients
ICCV	International Conference on Computer Vision
ID	Identifier
IEEE	Institute of Electrical and Electronics Engineers
IS	Individual Section (in glass forming)
ISBN	International Standard Book Number
LR	Learning Rate
MEEC	Master in Electrical and Computer Engineering
MICCAI	Medical Image Computing and Computer-Assisted Intervention
MIT	Massachusetts Institute of Technology
MST	Minimum Spanning Tree
NIPS	Neural Information Processing Systems
NMS	Non-Maximum Suppression
PCA	Principal Component Analysis
PSL	Pressure Sensitive Labelling
RGB	Red, Green, Blue
RPN	Region Proposal Network
SCP	Stones, Ceramics, and Porcelains
SDG	Sustainable Development Goals
SGD	Stochastic Gradient Descent

SSD	Single Shot MultiBox Detector
SVM	Support Vector Machine
TIP	IEEE Transactions on Image Processing
TN	True Negative
TP	True Positive
UN	United Nations
YOLO	You Only Look Once

Chapter 1

Introduction

1.1 Context

BA Glass is a national company with over a century of experience in the development, production, and commercialization of glass containers for the food, beverage, pharmaceutical, and cosmetic industries. The company manufactures and exports more than 10 billion bottles and jars annually, with the food and beverage sector representing its main customer base. This sector operates with continuous high-volume production, which requires frequent adjustments to production processes. As a result, the company prioritizes improving equipment availability and reliability, reducing production losses, and increasing overall operational efficiency.

The glass manufacturing process is complex and comprises several critical stages. Initially, raw materials are stored, weighed, and combined in specific proportions to form a batch mixture. This mixture is then melted and filtered. The resulting molten glass enters the conditioning stage, where it is processed to achieve thermal uniformity. This step is essential to meet the required quality standards in the final product. Following conditioning, the moulding stage gives the glass its final shape. A thermal treatment process is subsequently applied to relieve internal stresses generated during moulding. The containers then undergo quality control procedures, during which multiple inspection stages are conducted to detect any defects. Defective containers are automatically rejected and reintroduced into the production cycle as raw material.

Recycled glass, referred to as cullet, plays a central role in the glass manufacturing process. Its use presents both environmental and economic advantages. As a non-degradable, inert material, glass can be recycled indefinitely without loss of quality, supporting a closed-loop production model. The substitution of raw materials with cullet helps reduce CO₂ emissions, contributing to the industry's carbon mitigation goals[1].

In 2023, BA Glass reported a cullet utilization rate of 35.7% across its annual production. This figure represents a significant contribution to lowering the company's carbon footprint. With 2020 as a reference year, BA Glass has set specific environmental targets: a 50% reduction in CO₂ emissions by 2035 and the attainment of carbon neutrality by 2050[2].

1.2 Motivation

Cullet treatment plants are responsible for processing recycled glass obtained from a variety of sources. At BA Glass, this material includes residues collected from urban waste streams and other glass-related industries, resulting in batches with heterogeneous composition and quality. To prepare recycled glass for use in production, several treatment procedures are applied to remove contaminants and ensure compliance with quality standards.

Despite these procedures, the final cullet is not entirely free from impurities. Residual contaminants, such as ceramics, may remain even after mechanical sorting and filtering. To maintain product quality and avoid equipment damage, a maximum threshold of 50 grams of contaminants per tonne of cullet has been defined. Exceeding this limit can lead to significant defects in the final containers and disrupt the manufacturing process.

At the Avintes treatment facility, cullet quality control is currently conducted manually. Every three hours, a sample of approximately 10 kg is collected and visually inspected. Contaminants are manually sorted, weighed, and used to calculate the concentration of impurities in parts per million (ppm). This manual approach introduces uncertainty into the estimation process, both due to sampling limitations and potential human error in classification and measurement.

This thesis proposes a deep learning-based approach to automate the quality estimation of cullet samples. The objective is to remove human dependency in the identification, classification, and quantification of contaminants. By applying image-based analysis, the model is designed to detect and classify the particles in the sample, both recyclable glass and contaminants, according to pre-defined classes. Based on this classification, the expected weight distribution of the sample can be estimated to assess its quality.

In addition to reducing operator influence, the proposed approach increases the scalability and consistency of the process. Larger sample volumes could be processed in shorter time frames, providing more representative and reliable quality assessments across cullet batches.

1.3 Objectives

Given the potential errors in the quality estimation process and its critical importance, it is desirable to implement a system capable of accurately identifying contaminants and determining the quality of cullet samples.

This project is dedicated to addressing the challenges associated with automating the quality estimation of cullet. The work is structured around two primary objectives:

- **Image-Based Detection of Cullet Particles:** Develop and validate a deep learning architecture capable of identifying and classifying individual particles present in sample images. The model must be capable of distinguishing between recyclable glass and contaminants, assigning each object to a predefined material class.
- **Sample Quality Estimation:** Use the classification results to compute a quantitative measure of sample purity. This index reflects the estimated weight distribution across classes

and serves as a proxy for the quality of the sample. A higher proportion of contaminants corresponds to a lower sample quality.

1.4 Document Structure

The remainder of this document is organized into five main chapters:

- **Chapter 2 – Background:** Introduces the context and motivation for cullet quality assessment, presenting the relevant industrial processes and quality control requirements.
- **Chapter 3 – Literature Review:** Surveys existing approaches to object detection, classification, and image-based material analysis, highlighting the limitations and gaps that motivate the proposed methodology.
- **Chapter 4 – Methodology:** Describes the design and implementation of the proposed deep learning pipeline, including dataset creation, model architectures, and the mass estimation procedure.
- **Chapter 5 – Experimental Results:** Presents the evaluation of the pipeline through quantitative and qualitative results, assessing the performance of each module and the system as a whole.
- **Chapter 6 – Conclusions and Future Work:** Summarizes the key contributions and findings of the work, and outlines potential directions for further research and system improvement.

Chapter 2

Background - Scientific Principles and Manufacturing of Glass Containers

Glass is a material that can be naturally formed in the nature. However, due to the demand for glass exceeding its natural availability, extensive research has been conducted on synthetic glass production.

In the context of this dissertation, for contextual purposes, the following sections will present overview of the glass formation process (Section 2.1) and details regarding the glass industry and its manufacturing processes (Section 2.2).

2.1 Glass Fundamentals - Definition and Properties of Glass

The majority of synthetically produced glass is primarily composed of silica. The process generally involves combining and melting raw materials, followed by controlled cooling of the resulting mixture. However, neither the manufacturing process nor the chemical composition alone is sufficient to fully characterize this material. For a material to be classified as glass, it must exhibit the following characteristics:

- **Time-dependent glass transformation behaviour:** A defining feature of glasses is their departure from thermodynamic equilibrium upon cooling. This behaviour is best illustrated using a volume–temperature diagram, as shown in Figure 2.1. When a liquid is cooled slowly from an initial high-temperature state (point A), its volume decreases steadily along the path A→B. If crystallization occurs—typically at the freezing temperature (T_f)—a sharp drop in volume is observed (path B→C), followed by further contraction along C→D as a crystalline solid.

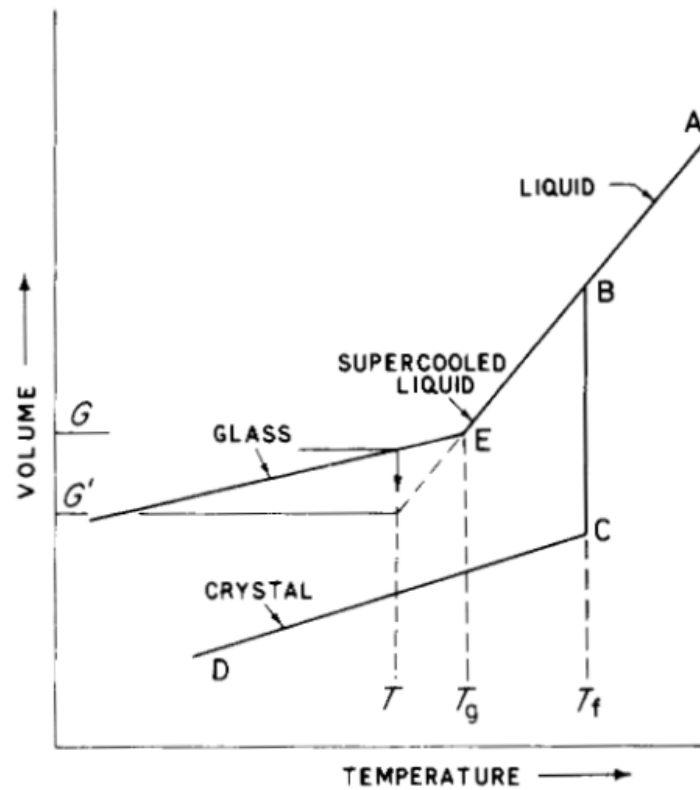


Figure 2.1: Schematic representation of the volume–temperature relationship for a liquid, supercooled liquid, glass, and crystalline solid. Slow cooling enables crystallization (path $AB \rightarrow C \rightarrow D$), while rapid cooling avoids nucleation and leads to glass formation (path $AB \rightarrow E \rightarrow G$). The transformation range (T_g) marks the point where structural relaxation times exceed the experimental timescale. The fictive temperature (T_f) denotes the temperature at which the glass structure reflects the state of the supercooled liquid. The continued decrease in volume from G to G' (at constant temperature T) exemplifies stabilization. Adapted from [3].

However, if the cooling rate is sufficiently high to bypass crystallization, the system enters a supercooled liquid state and follows a continuous contraction path ($B \rightarrow E$). Upon further cooling, the system reaches a temperature T_g , known as the glass transition temperature and the system falls out of equilibrium. This results in the formation of a rigid, disordered solid—a glass—along the path $E \rightarrow G$.

Notably, the position of the transition point T_g (and thus point E) is not fixed but varies with the cooling rate. Faster cooling leads to higher T_g values, while slower cooling shifts it lower. Therefore, it is more accurate to describe the transformation as occurring over a *glass transition range* rather than at a precise temperature.

According to the formal definition provided by Shelby, “Any material, inorganic, organic, or metallic, formed by any technique, which exhibits glass transformation behaviour is a glass” [4]. This definition emphasizes that the glassy state is not determined solely by composition or processing, but by its dynamic response to cooling—manifested through a

non-equilibrium, time-dependent transformation into a rigid state. This kinetic viewpoint is now widely accepted in glass science literature, and is foundational to understanding relaxation, aging, and thermal history effects in glasses [3], [5].

- **Absence of long-range atomic order:** The fundamental distinction between glasses and crystalline solids lies in their atomic-scale organization. Crystals exhibit a highly ordered, periodic arrangement of atoms, resulting in long-range translational symmetry. In contrast, glasses lack such periodicity; their atomic arrangement is disordered beyond short-range correlations, leading to the absence of long-range order [6]. This structural contrast is illustrated in Figure 2.2, where the difference between crystalline and glassy phases is depicted schematically.

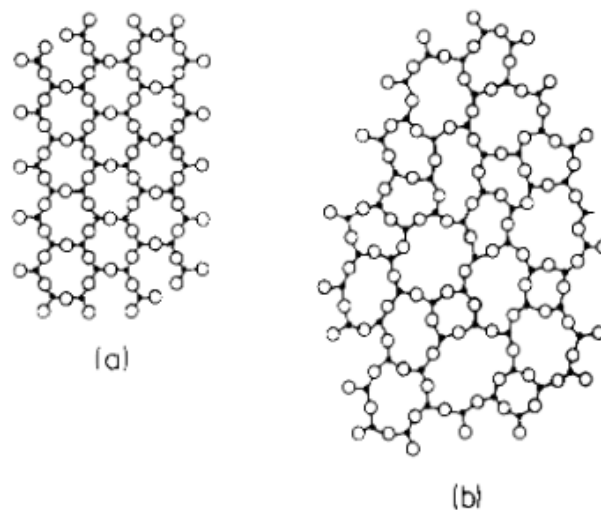


Figure 2.2: Two-dimensional schematic of (a) a crystalline structure with long-range atomic order and (b) the disordered glassy form of the same compound. Adapted from [7].

2.2 Glass Containers Manufacturing

Early synthetic glass creation was first fashioned into beads and jewellery. Over time, advances in glass-working techniques led to the development of moulding methods that enabled the production of glass containers [7]. Under Roman influence, glass vessels became widespread across the Mediterranean and, from that on, the use of coloured glass also expanded, with methods for producing various hues often closely guarded. Later, the introduction of glass-blowing further advanced the craft [8].

Industrialization in the late 19th and early 20th centuries created sustained demand from food and beverage markets. The invention of the first fully automatic bottle-making machine, by Michael J. Owens in 1903, marked a key turning point, significantly improving production speed and uniformity [9]. Further advancements included the Individual Section (IS) machine, which

allowed independent operation of forming cavities, enhancing manufacturing flexibility and efficiency. Additionally, improvements in annealing, mould materials, computerized control and automated inspection machines have enabled real-time defect detection and closed-loop optimization [10].

In recent decades, sustainable production planning has become increasingly important. The use of recycled glass as a substitute for carbon-intensive raw materials is now a common practice to reduce energy consumption and CO₂ emissions. However, residual contaminants and colour carry-over can propagate into the finished containers if not rigorously controlled [1]. These defects may lead to the rejection of affected containers, reducing overall production efficiency. Therefore, the development of quality control procedures for cullet is essential to stabilise glass container manufacturing.

The following sections present the container morphology and typologies (Section 2.2.1), describe the manufacturing process (Section 2.2.2), explain the integration of recycled glass (Section 2.2.3), and summarise the main defect categories (Section 2.2.4), establishing the technical framework for the automated cullet quality assessment addressed in this thesis.

2.2.1 Glass Container Morphology

The composition of a glass container can be divided in several distinct regions, each designed to fulfill specific functional and mechanical roles throughout the container's lifecycle. The principal morphological elements are identified in the Figure 2.3 and described as follows:

- **Finish:** Located at the topmost portion of the container, the finish wraps the sealing surface, bore (or mouth), and the bead or collar (ring). The sealing surface ensures compatibility with closure systems (e.g., caps, corks), while the bore enables product filling and dispensing. The collar acts as mechanical support when transferring the parison into the blow mould during forming.
- **Neck:** This region serves as connection from the collar to the top-end of the shoulder. This structural part strengthens the container structural integrity to withstand compressive loads during sealing.
- **Shoulder:** The shoulder is the curved transition between the neck and the body. Its geometry influences the flow behaviour during filling and can affect the stress distribution under internal pressure or impact.
- **Body:** The central and typically largest section of the container, the body is the primary storage volume. It is designed to maximize capacity while maintaining mechanical stability.
- **InswEEP (Heel):** The insweep is the curved region connecting the body to the bottom. This part design optimizes glass distribution during the forming process and improves the structural integrity of the container by reducing stress concentrations at the junction.

- **Bottom:** This is the lowermost portion of the container that supports the entire structure in an upright position. It must be geometrically stable and flat enough to ensure steady placement on surfaces.
- **Push-Up:** Located at the center of the bottom, the push-up is an elevated dome that raises the central area of the base above the resting surface. This feature enhances container stability, facilitates proper stacking, and reduces the likelihood of contact-induced thermal shock or impact damage to the bottom.

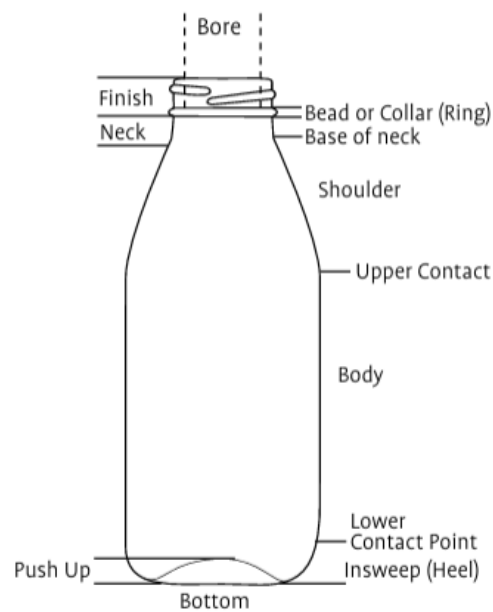


Figure 2.3: Schematic Glass Container Composition. Adapted from [11].

Container Typologies

Glass containers are produced in a variety of forms to meet diverse functional, aesthetic, and commercial requirements. The main morphological differences are in the mouth configuration, body shape (round or non-round), and neck profile (wide or narrow).

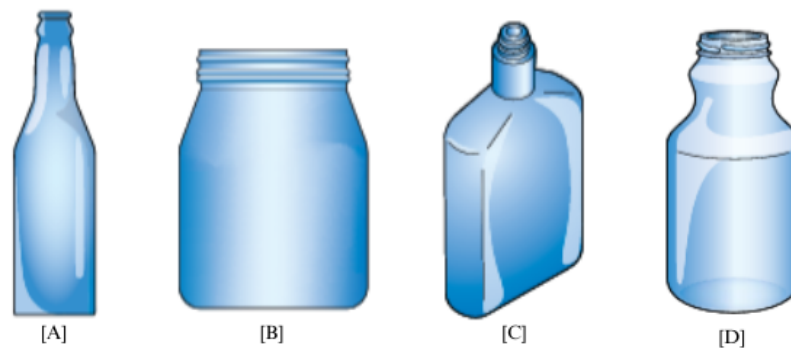


Figure 2.4: Illustration of common glass container types. [A] Narrow-neck bottle; [B] Wide-mouth jar; [C] Non-round bottle (irregular profile); [D] Round non-cylindrical bottle. Adapted from [11].

Figure 2.4 illustrates four representative container types commonly encountered in industrial production:

- **[A] Narrow-neck bottle:** Typically used for beverages and pharmaceuticals. These containers require blow-and-blow forming techniques due to the restricted finish geometry, where internal pressure is used to form the parison and final shape [9], [11].
- **[B] Wide-mouth jar:** Commonly found in food packaging applications (e.g., jams, sauces). The press-and-blow process is preferred, as it facilitates uniform wall thickness and precise control of mouth dimensions [10].
- **[C] Non-round bottle:** Irregular cross-sectional profiles increase mould design complexity and require precise thermal and mechanical control to ensure uniform glass distribution and dimensional stability [11].
- **[D] Round non-cylindrical bottle:** These containers retain overall rotational symmetry but exhibit variations in diameter or curvature along the body. Such features influence cooling rates, parison formation, and stress development during annealing [7].

The design differences influence the production process requiring tailored conditions, including specific gob dimensions, mould surface treatments, and forming cycles. Consequently, container geometry is a fundamental variable that directly impacts machine setup, defect prevention, and production celerity [9], [11].

2.2.2 Glass Containers Manufacturing Process

The production of glass containers is a sophisticated, multi-stage industrial process that transforms raw materials into precise and durable packaging products. This highly automated sequence encompasses raw material preparation, melting, forming, annealing, inspection, and final packaging. The main steps are illustrated in Figure 2.5.

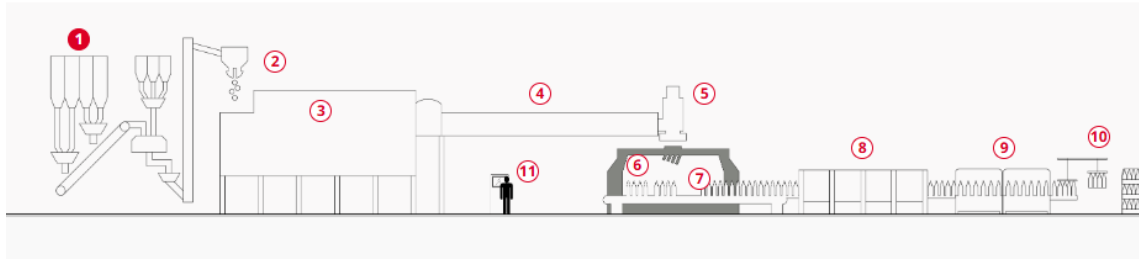


Figure 2.5: Schematic of the glass container production line. Adapted from [12].

Batch Preparation and Raw Material Handling

The process begins at the *batch plant* (1), where raw materials—including quartz sand (SiO_2), soda ash (Na_2CO_3), limestone (CaCO_3), dolomite, and feldspar—are stored, mixed, and weighed in carefully controlled proportions. These components form the basis of soda-lime-silica glass, the most common composition for containers [4], [7]. Additives such as iron oxide, cobalt, or sulphur compounds may also be introduced to impart colour.

Charging and Melting

The homogenized batch is conveyed to the *batch charger* (2) where it is continuously introduced into the *furnace* (3). Within the furnace, the batch melts at temperatures between 1500 and 1600 °C. Heat is recovered using regenerative or recuperative systems to improve thermal efficiency [3]. The melting process lasts approximately 24 hours to ensure complete homogenization.

Conditioning and Gob Formation

The molten glass flows to the *forehearth* (4), a temperature-controlled channel that ensures thermal uniformity before forming. From there, the glass enters the *gob formation* system (5), where a plunger-and-shear mechanism cuts the melt into precise gobs. These gobs are the input for the forming stage.

Container Forming and Moulding

Forming occurs in the *IS machine* (6), which contains multiple independent sections, each equipped with moulding cavities. Two dominant forming processes are used:

- **Blow-and-blow process:** Compressed air forms a preliminary shape (parison) and then expands it into the final form. Suitable for narrow-neck containers.

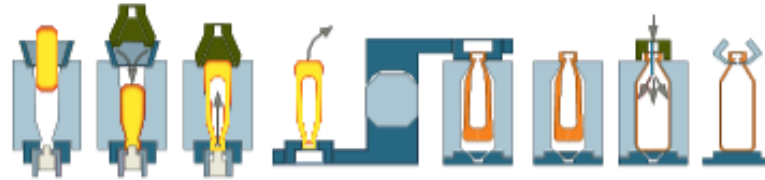


Figure 2.6: Schematic of the Blow-and-blow process. Adapted from [11].

- **Press-and-blow process:** A plunger creates the parison, followed by final blowing. Common for wide-mouth jars.



Figure 2.7: Schematic of the Press-and-blow process. Adapted from [11].

Each section of the *IS machine* can simultaneously produce several containers, ensuring high throughput and modular control [9].

Take-Out, Coating, and Annealing

Freshly moulded containers, still at temperatures around 600 °C, are transferred using a *take-out system* (7) to the conveyor. To prevent surface damage, a thin coating of tin or titanium chloride is applied at the *Hot End*, forming a protective layer. Containers then enter the *annealing lehr* (8), where they are reheated to around 580–650 °C and gradually cooled to ambient temperature. This controlled thermal cycle relieves internal stresses and ensures dimensional stability [10].

At the *Cold End*, an additional polyethylene-based coating may be applied to enhance mechanical resistance and prevent abrasion during handling.

Inspection and Quality Control

After annealing, containers are subjected to rigorous inspection (9). Automated systems verify dimensional tolerances, surface integrity, and internal cleanliness using cameras, lasers, and mechanical gauges. Defective products are rejected and recycled back into the batch as cullet.

Packaging and Customization

Accepted containers proceed to the *packaging zone* (10), where they are palletized and shrink-wrapped for shipment. Optionally, containers may undergo customization through painting, pressure-sensitive labelling (PSL), sleeving, engraving, or screen printing (serigraphy), depending on customer specifications.

Automation and Control Systems

The entire production line is monitored and controlled by integrated automation platforms (11), enabling real-time supervision of temperature profiles, flow rates, pressure curves, and defect detection rates. These systems enhance productivity and minimize waste through predictive maintenance and closed-loop feedback control [9], [10].

2.2.3 Integration of Recycled Glass in Container Manufacturing

The use of recycled glass is a crucial practice in glass manufacturing, offering substantial environmental and economic benefits. Glass, being a durable and inert material, can be recycled indefinitely without compromising its intrinsic properties, thereby enabling a highly efficient closed-loop system. Cullet is essential to this sector since it directly replaces carbon-intensive raw materials, which significantly reduces energy consumption and carbon emissions [1].

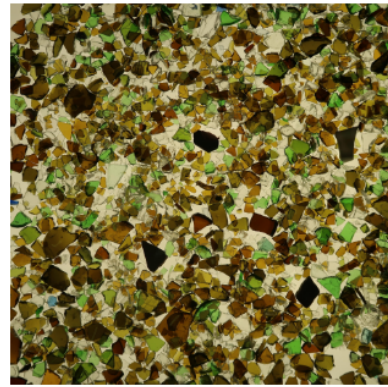
The Cullet Treatment Plants receive and process recycled glass from various sources, acquiring residues from urban waste collection and other glass-related industries. As a result, the incoming material is typically heterogeneous in both composition and quality. The recycled material is processed and, the purified output referred to as cullet, is used as a raw material in glass manufacturing.

The chemical and physical characteristics of the cullet, including its colour composition and the presence of residual contaminants, directly influence the properties of the final glass product. In particular, the colour of the recycled glass plays a critical role in determining the appearance of newly manufactured containers. Even small quantities of coloured glass, such as green or amber fragments, can affect the composition of the gob leading to unwanted colouration.

To address this, cullet treatment plants typically separate the output into two main categories. The *flint cullet*, composed primarily of colourless transparent glass, and *mixed cullet*, consisting of a mixture of different coloured glass. Flint cullet is used in the production of transparent glass containers, where colour contamination must be minimized. Mixed cullet, by contrast, is used in the production of coloured glass containers. The Figure 2.8 presents examples of both cullet types.



(a) Flint cullet: composed mainly of transparent glass fragments.



(b) Mixed cullet: composed of green, amber, and other coloured glass fragments.

Figure 2.8: Examples of cullet types sorted during the treatment process.

Cullet Purification and Preparation for Glass Production

The transformation of recycled glass into cullet suitable for glass production involves a sequence of industrial operations aimed at contaminant removal and achieving uniform particle size. Figure 2.9 presents the main stages involved in converting recycled glass into cullet for container glass manufacturing. A more detailed version of this process is shown in Figure A.1. However, for the purposes of this study, only the main stages of the transformation process are considered. Detailed procedural steps are beyond the scope of this work and can be referenced in the aforementioned flow-chart.

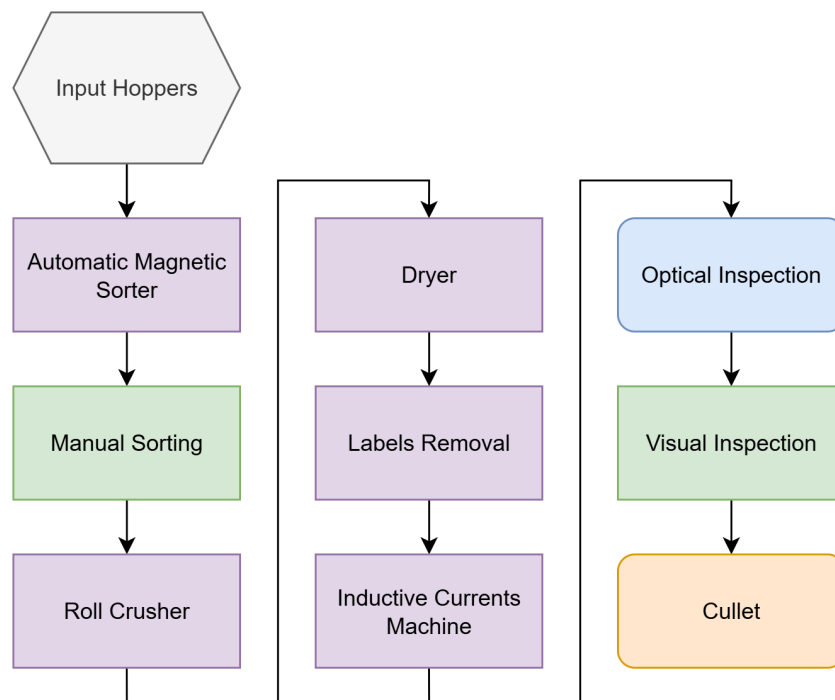


Figure 2.9: Schematic of the process to treat recycled glass

The process begins with the introduction of untreated recycled glass into input hoppers using wheel loaders. The batches of material contain various contaminants mixed with the glass. To minimize the impact of highly contaminated batches, material is fed from multiple hoppers at different flow rates. This approach helps to dilute lower-quality, highly contaminated, inputs and stabilize the overall composition of the recycled glass stream.

The first separation step targets ferrous metals. Iron-containing contaminants are eliminated from the glass stream by an automated magnetic separation. This is followed by a manual sorting stage, where operators extract contaminants such as stones, ceramics and plastics. The primary objective is to eliminate pollutants with larger dimensions that could disrupt the operation of downstream equipment or possibly harm or obstruct certain sorting stations.

The recycled material is subsequently fed into a roll crusher. In order to transform the material flow into a uniform batch, this equipment is used to crush the particles with dimensions above a certain threshold. The resultant mixture is conducted to the drying system. During this stage, the recycled glass is cleaned of any remainings of moisture or dust that can interfere with label removal, optical sorting, or other mechanical separation processes.

After drying, the next step involves the removal of surface labels, which may still be adhered to glass fragments. This is carried out by an automated delabeling system, in which the material is agitated inside a chamber equipped with rotor and stator blades. The mechanical interaction between the blades and the glass particles detaches the labels, which are then separated from the glass stream using a directed airflow. Effective label removal is essential to prevent interference with subsequent optical sorting and to enhance the accuracy of contaminant detection.

Following this, non-ferrous metals such as aluminium, copper, and brass are removed using an eddy current separator. This machine uses a high-speed rotating magnetic rotor to induce eddy currents in conductive non-ferrous metals, generating repulsive forces that separate them from the non-conductive glass material.

Once metallic contaminants have been removed, the cullet stream undergoes optical sorting. A set of automated sorting units, each calibrated to a specific particle size fraction, use machine vision systems to identify and remove non-glass materials based on differences in optical properties.

The final step in the cullet treatment process is a manual visual inspection. This stage serves as a quality control measure to identify and remove any remaining contaminants that were not detected or eliminated by the automated systems. It also provides critical feedback on the performance of the upstream equipment. For instance, the appearance of a localized concentration of contaminants, referred to as a “contamination line”, may indicate a malfunction in one of the optical sorting units, such as a defective sensor or actuator, which fails to properly eject unwanted material at a specific point on the conveyor. Once the material passes this final inspection, the treated cullet is either stored in silos or transported to glass manufacturing facilities, where it is used as a raw material in the production of new glass containers.

Quality Control Analysis for Cullet

The purification process applied to recycled glass in treatment facilities is not entirely effective. As result, the cullet, often contains residual contaminants such as ceramics or stones. When introduced into the furnace, these materials can affect the chemical and physical consistency of the molten glass. During the forming stage, such contamination may give rise to defects in the final containers. Section 2.2.4 provides an overview of common defects and their causes. Specifically, *Material or Composition Defects* are frequently associated with the use of low-quality cullet in batch formulation.

To mitigate this, quality control mechanisms are implemented to assess the cullet composition and reject substandard material at the source. The control is based on periodical checks, under the assumption that each sample is representative of the cullet processed during the time interval between inspections. A sample of approximately 10 kg is collected and examined through manual separation of contaminants, followed by weighing and calculating the contaminant content in parts per million (ppm). The analysis also quantifies the presence of glass by colour and determines the respective proportions of each category. The Table 2.1 presents the results of this analysis for five samples of flint.

Table 2.1: Flint Cullet Component Analysis

(a) Absolute weights of cullet components

N°	Sample Weight (kg)	White (kg)	Blue (g)	Green Amber (g)	Other Whites (g)	SCP (g)	Opals (g)	Plastics (g)	Metals (g)
1	10.52	10.25	201.42	31.04	39.77	0	0	0	0
2	10.41	10.21	143.14	16.94	37.90	0	0	0	0
3	10.47	10.25	166.51	25.39	32.39	0	0	0	0
4	10.40	10.18	163.88	21.49	37.14	0	0	0	0
5	10.57	10.32	171.33	44.84	38.20	0	0	0	0

(b) Relative composition of cullet

N°	White (%)	Blue (%)	Other Whites (%)	Green Amber (%)	SCP (%)	Opals (%)	Plastics (%)	Metals (%)
1	97.43	1.91	0.30	0.38	0	0	0	0
2	98.17	1.38	0.16	0.36	0	0	0	0
3	97.81	1.59	0.24	0.31	0	0	0	0
4	97.88	1.58	0.21	0.36	0	0	0	0
5	97.54	1.62	0.42	0.36	0	0	0	0

As previously stated, the cullet quality analysis procedure involves the manual sorting and weighing of particles according to predefined material classes. Figure 2.10 presents representative visual examples of typical materials associated with each class. These classes are described as follows:

- **White Glass:** Recycled glass fragments without colouration. This class represents the target material in flint cullet processing. It typically constitutes the majority of the sample, as treatment processes are designed to extract and preserve this type of glass.
- **Blue Glass:** Recycled glass particles exhibiting a blue colouration. These fragments are commonly derived from glass containers such as beverage bottles.
- **Other Whites Glass:** Recycled glass fragments with unclear or mixed colouration that do not fit the other colour categories. These particles are often gray in appearance, tend to have significantly larger dimensions, and are generally thicker than typical white glass fragments.

- **Green Amber Glass:** Coloured glass fragments in shades of green or amber. This class includes particles primarily originating from wine and beer bottles and is typically the most prevalent type of glass received at treatment facilities.
- **SCP (Stones, Ceramics, and Porcelains):** Non-glass inclusions composed of mineral-based materials. These contaminants are introduced through improper sorting, where items such as ceramic tableware or porcelain fragments are mistakenly discarded with glass waste. They may also originate from debris generated during storage and transport.
- **Opals:** Opaque mineral fragments, typically white but possibly varying in colour from milky blue to gray. Similar in origin to SCP, these materials are commonly introduced through cross-contamination during transport or public disposal errors.
- **Plastics:** Synthetic polymer contaminants of various shapes and colours. Typical examples include caps, pourers, and other plastic components associated with glass containers. These fragments often persist due to incomplete separation in preprocessing stages.
- **Metals:** Metallic contaminants generally associated with closures, caps (e.g., beer bottle tops), or decorative elements attached to the original glass containers. These materials are introduced during collection and are not always removed during initial sorting.

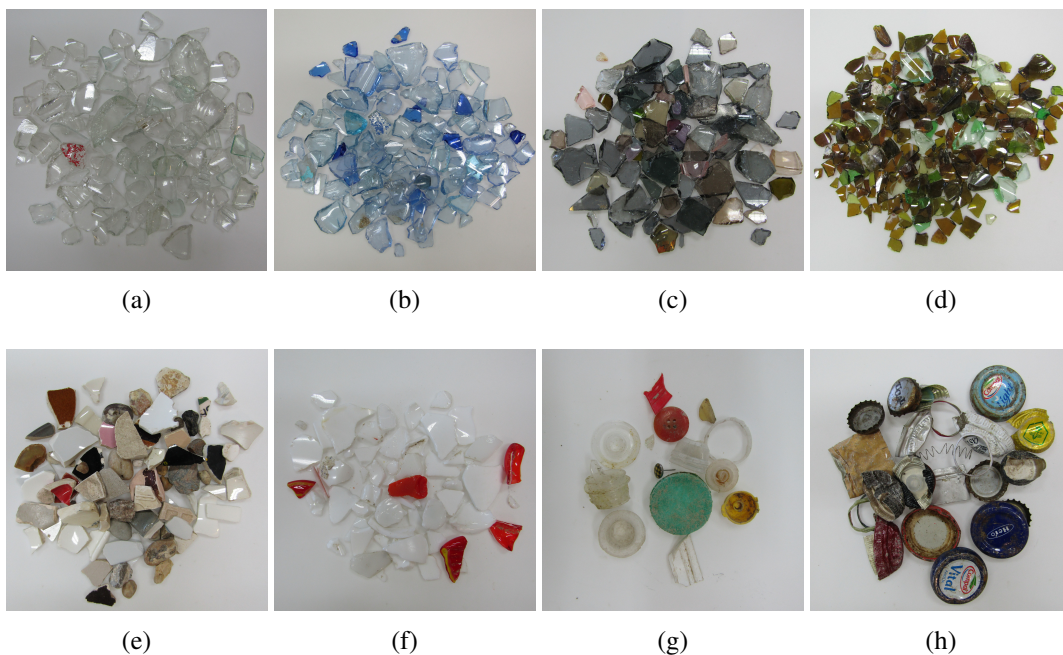


Figure 2.10: Representative examples of components identified in a flint cullet sample. Each sub-figure illustrates a distinct material class consistent with the cullet component categories defined in Table 2.1: (a) White Glass, (b) Blue Glass, (c) Other Whites Glass, (d) Green Amber Glass, (e) SCP (Stones, Ceramics, and Porcelains), (f) Opals, (g) Plastics, and (h) Metals.

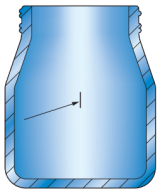
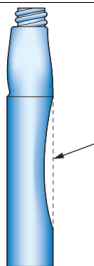
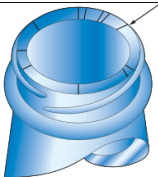
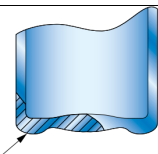
2.2.4 Glass Containers Defects

Defects that arise in the production of glass containers can be more effectively analysed when categorized by their root causes or operational consequences, rather than solely by location. The following tables summarize the most common defect types, their typical causes, recommended corrective actions, and provide space for illustrative examples.

Thermal Stress Defects

Thermal stress defects occur when temperature variations within the container are not properly controlled during forming, annealing, or cooling. These temperature gradients cause internal causing stress and structural weaknesses in the container and lead to cracks or deformation.

Table 2.2: Representative examples of thermal stress defects in glass containers, including their causes, corrective strategies, and visual illustrations. Adapted from [11].

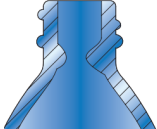
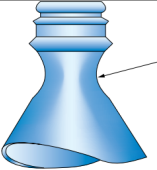
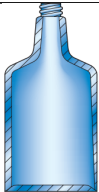
Defect	Cause Summary	Key Remedies	Illustration
Pressure Checks	Overcooling, excessive blow pressure, poor mould venting	Adjust blow parameters; improve mould venting	
Sunken Sides	Inadequate cooling; poor mould contact	Increase mould cooling; optimize blow pressure	
Crizzled Finish	Rapid cooling, resulting in surface stress	Controlled annealing; reduce cooling gradients	
Bottom or Base Checks	Uneven cooling or inconsistent contact	Adjust annealing conditions; refine mould design	

Mitigating thermal stress defects requires carefully calibrated annealing schedules, consistent mould temperature control, and precise synchronization of forming and cooling operations. Preventive maintenance and thermal profiling are essential for process stability and container reliability.

Mechanical Forming Defects

Defects in mechanical forming usually arise from problems with the functioning or state of the forming machinery. Common sources of these faults include irregular gob geometry, misaligned, damaged, or mismatched mould components, and improper timing of forming cycles. The impact is often dimensional in nature, leading to deformities that compromise the appearance and functionality of the container.

Table 2.3: Representative examples of mechanical forming defects in glass containers, including their causes, corrective strategies, and visual illustrations. Adapted from [11].

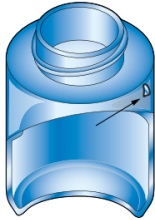
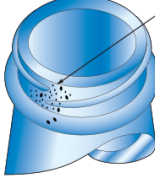
Defect	Cause Summary	Key Remedies	Illustration
Bent Finish	Improper neck ring closure; misaligned equipment	Adjust timing; calibrate mould alignment	
Choked Neck	Incomplete plunger movement; misdirected gob	Check gob cut; adjust plunger positioning	
Pinched Neck	Premature mold closure; poor alignment of parts	Adjust mould closing sequence; replace worn components	
Thin Ware	Uneven glass distribution during forming	Optimize parison formation; adjust blank mould settings	

Prevention of mechanical forming defects requires precise maintenance and calibration of forming components, consistent gob geometry, and careful synchronization of forming cycle phases. Regular inspection and modular replacement of mould parts help ensure reliable and repeatable production quality.

Surface Defects from Contamination or Wear

Surface defects are typically superficial imperfections that appear on the exterior of the glass container as a result of contamination or mechanical wear during the hot-end forming process. While they may not always compromise the structural integrity of the container, these defects degrade optical clarity and surface smoothness. These defects are often introduced by improper maintenance, excessive lubrication, or contaminated mould surfaces.

Table 2.4: Representative examples of surface defects from contamination or wear in glass containers, including their causes, corrective strategies, and visual illustrations. Adapted from [11].

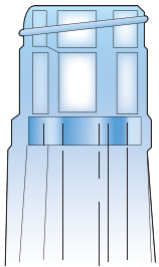
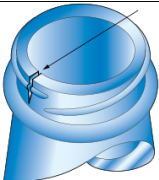
Defect	Cause Summary	Key Remedies	Illustration
Stuck Glass Particles	Contamination from shearing/swatching; inadequate cleaning	Replace or recalibrate shears; clean moulds thoroughly	
Oil Marks	Excessive lubrication; residue from swabbing	Adjust lubrication levels; improve equipment cleaning	
Dirty or Rough Finish	Contaminated moulds; improper swabbing techniques	Clean or replace mold parts; use correct swab compounds	

To mitigate surface defects, strict hygiene practices must be applied in mould maintenance, lubrication procedures, and ambient dust control. Routine inspection and cleaning of forming equipment are essential to preserving surface quality and maintaining high production yields.

Structural or Dimensional Defects

Structural or dimensional defects arise when a container deviates from the geometrical specifications defined by product design or industry standards. These deviations often occur due to problems during the forming process—such as equipment misalignment, improper mould design, or instability during cooling—and typically lead to containers that do not meet tolerance limits in shape, symmetry, wall thickness, or sealing geometry.

Table 2.5: Representative examples of structural or dimensional defects in glass containers, including their causes, corrective strategies, and visual illustrations. Adapted from [11].

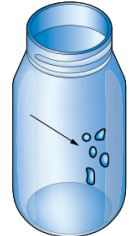
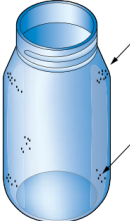
Defect	Cause Summary	Key Remedies	Illustration
Offset/Out-of-Round Finish	Misaligned molds or uneven cooling stresses	Re-calibrate mould alignment; optimize thermal profiles	
Out-of-Shape Ware (Leaner)	Improper base support; dimensional imbalances during forming	Adjust machine settings; ensure uniform cooling	
Split Finish	Excessive mechanical force or improper shear action	Modify shear timing; regulate gob temperature; check equipment alignment	

These defects may not always result in complete functional failure, they often compromise container performance in automated filling, sealing, and packaging lines. Moreover, visual asymmetries or poor dimensional control can result in customer rejection due to perceived or actual quality issues.

Material or Composition Defects

Material or composition defects originate primarily from inconsistencies in the batch formulation or contamination introduced during melting and gob formation. These defects compromise the purity and uniformity of the final glass container.

Table 2.6: Representative examples of material or composition defects in glass containers, including their causes, corrective strategies, and visual illustrations. Adapted from [11].

Defect	Cause Summary	Key Remedies	Illustration
Cords	Segregation or inhomogeneity in the batch mix	Verify and adjust batch composition; ensure thorough mixing	
Blisters/Seeds	Gas entrapment during melting; improper refining	Improve furnace control; enhance batch homogenization	
Black Spots	Inclusions from refractory material or contaminated cullet	Improve raw material screening; upgrade purification processes	

Effective prevention of material and composition defects requires stringent raw material screening, consistent batch preparation, optimized melting and refining processes, and rigorous equipment maintenance throughout the hot-end production stages.

Chapter 3

Literature Review

The present chapter surveys the key methodologies relevant to this dissertation’s technical scope, with a focus on computer vision approaches applied to image understanding and object recognition. The selected literature provides the foundational understanding necessary to develop and evaluate modern object detection and segmentation systems.

To contextualize the model selection and methodological framework used in this work, the following sections provide an overview of 2D object detection techniques (Section 3.1) and 2D image segmentation methodologies (Section 3.2), highlighting the evolution from classical algorithms to deep learning-based paradigms. These sections emphasize both theoretical principles and practical architectures, establishing a foundation for the subsequent system implementation and analysis.

3.1 2D Object Detection Methodologies

Object detection is a computer vision task that focuses on identifying and locating instances of visual objects belonging to a specific class within digital images. Unlike image classification, which aims to understand and categorize an image as a whole under a single label.

Performs both classification and localization of multiple objects within an image. This is achieved by identifying the object classes and drawing bounding boxes around the detected instances. This method extends classification capabilities by providing spatial information about each object in an image, making it a crucial component in applications such as autonomous vehicles, surveillance, and robotics.

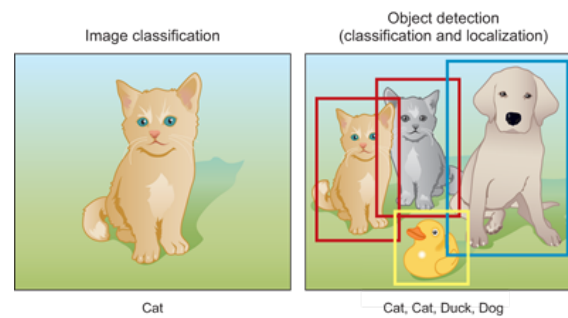


Figure 3.1: Comparison between image classification and object detection. Image classification identifies the primary object in an image ('Cat'). Object detection performs both classification and localization, identifying multiple objects (e.g., 'Cat,' 'Duck,' and 'Dog') and their positions within the image using bounding boxes. Figure adapted from [13].

Object detection has received considerable attention in recent years, driven by rapid technological advancements that have significantly shaped the field of computer vision. Its evolution can be divided into two distinct historical phases: the "traditional object detection period" (pre-2014) and the "deep learning-based detection period" (post-2014). These phases represent a transition from classical algorithmic approaches to deep learning-driven techniques that have revolutionized object detection and its applications [14].

3.1.1 Traditional Detectors

- Viola Jones Detectors:** In 2001, P. Viola and M. Jones revolutionized object detection by achieving real-time, unconstrained human face detection for the first time. The Viola-Jones framework introduced a novel approach to image classification, utilizing simple features inspired by Haar basis functions. Instead of relying directly on pixel intensities, the algorithm is able to calculate the sum of pixel intensities within rectangular regions. These values are used as features providing a more abstract and effective representation of image data. To compute the features rapidly, the framework uses the Integral Image, a data structure that allows for the efficient computation of rectangular sums. The most critical features are selected using a modified AdaBoost algorithm, which iteratively combines weak classifiers to build a strong classifier. This combination of fast feature computation and AdaBoost-based feature selection empowers the Viola-Jones framework to perform robust and real-time object detection with high accuracy [15].
- HOG Detector:** The method proposed by Dalal and Triggs for object detection relies on Histograms of Oriented Gradients (HOG), a feature descriptor that captures the local shape and appearance of objects in images. The process begins by computing image gradients to highlight edges and structures, followed by normalizing these gradients to reduce the effects of lighting and contrast variations. The image is divided into small regions called cells, where gradient orientations are summarized in histograms. These cells are then grouped into larger regions called blocks, which are normalized independently to enhance robustness to

lighting and contrast changes. The histograms from all blocks are combined into a single feature vector, representing the detection window. This feature vector is passed to a linear Support Vector Machine (SVM), which classifies the window as containing the target object (e.g., a person) or background. The detection window is scanned across the image at multiple positions and scales to detect objects of various sizes. Finally, overlapping detections are refined using non-maximum suppression to retain the most confident results, producing distinct bounding boxes for detected objects [16].

3.1.2 CNN-based Architectures

Convolutional Neural Networks (CNNs) are a specific type of Artificial Neural Networks (ANNs) designed for image recognition and processing tasks. Inspired by the biological processes of the visual cortex in animals, CNNs consist of neurons with learnable weights and biases. CNNs are composed of three primary types of layers, each serving a unique role in processing and extracting features from input images.

- **Convolutional Layer:** Performs convolution operations using kernels (or filters) to detect features such as edges and shapes. This process is made using a sliding window across the width and height of the image, computing dot products between the filter and the input values to produce activation maps.
- **Pooling Layer:** Performs downsampling operation typically applied after a convolutional layer. The activation maps are fed into pooling layers to downsample the data and reduce its dimensionality. This step enhances the model's efficiency and robustness by retaining essential features while discarding redundant information. Common pooling techniques include:
 - **Max pooling:** Retains the maximum value in each region.
 - **Average pooling:** Computes the average value in each region.
- **Fully-Connected Layer:** The final fully-connected layer takes the flattened and processed feature map and classifies the input image into predefined categories.

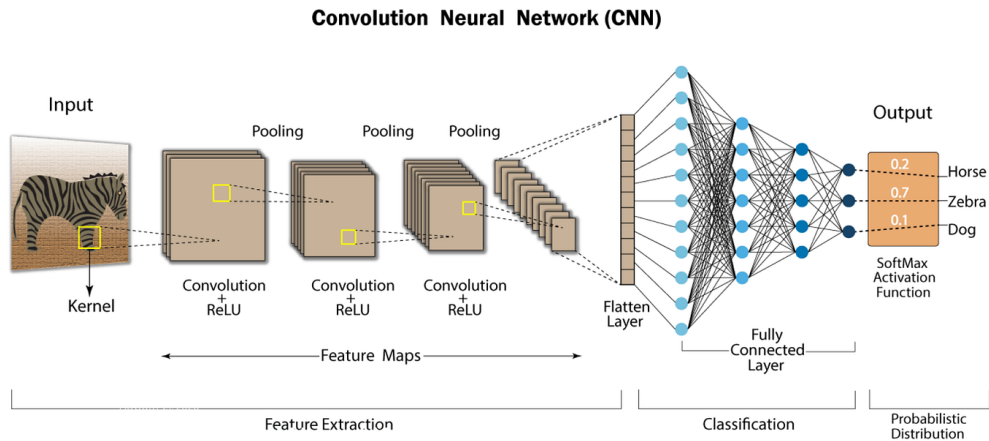


Figure 3.2: Architecture of a Convolutional Neural Network (CNN). The input image is processed through multiple convolutional layers with ReLU activation functions, followed by pooling layers that reduce dimensionality. The feature maps are flattened and passed through fully connected layers to perform classification. The output layer uses the SoftMax activation function to produce a probabilistic distribution over the possible classes, such as 'Horse,' 'Zebra,' or 'Dog'. Figure adapted from [17].

CNNs process images in a hierarchical manner, progressively extracting features at different levels. Initial layers detect simple patterns such as edges, corners, and textures and the deeper layers recognize more complex structures and prominent elements, such as shapes or specific objects.

Region Based Detectors

- **R-CNN:** The system consists of three main modules [18]:
 - **Region Proposals:** The system generates category-independent region proposals to define potential candidate detections. In the 2014 system proposed by R. Girshick, J. Donahue, T. Darrell, and J. Malik, selective search is utilized to enable controlled comparisons with previous detection methods.
 - **Feature Extraction:** Each region proposal is processed CNN to extract a 4096-dimensional feature vector. Features are computed by forward propagating a mean-subtracted 227×227 RGB image through five convolutional layers and two fully connected layers. To make region proposals compatible with the CNN's fixed input size (227×227 pixels), the regions are warped to the required dimensions. Padding is added around the bounding box to include contextual information.
 - **Classification:** The extracted feature vectors are fed into class-specific linear SVMs, which classify each region proposal into object categories.
- **Fast R-CNN:** Fast R-CNN is a object detection system that builds on the R-CNN framework, addressing its inefficiencies while improving speed and accuracy. Like its predecessor, the system consists in three main modules [19]:

- **Input and Feature Extraction:** The entire input image is processed by a deep CNN to generate a shared convolutional feature map. This process eliminates the need to extract features for each region proposal independently, as was done in R-CNN.
 - **Region of Interest (RoI) Pooling:** Instead of cropping regions from the original image, Fast R-CNN uses a RoI pooling layer to extract a fixed-length feature vector from the shared feature map for each region proposal. The pooling layer divides each region proposal into a fixed grid and applies max-pooling to ensure compatibility with the fully connected input layer.
 - **Classification and Bounding Box Regression:** The fixed-length feature vector for each proposal is processed through fully connected layers with two distinct heads: one for classification, which assigns a category label or background to the proposal, and another for regression, which computes the bounding boxes for the detected objects.
- **Faster R-CNN:** Faster R-CNN revolutionizes object detection by integrating region proposal generation directly into the neural network through the Region Proposal Network (RPN), eliminating the need for external methods like Selective Search. This innovation significantly enhances speed, efficiency, and accuracy compared to Fast R-CNN, establishing Faster R-CNN as a state-of-the-art object detection system. The RPN achieves real-time or near-real-time performance with improved accuracy [20].

One-Stage Detectors

- **You Only Look Once (YOLO):** YOLO approaches object detection as a single regression problem. Unlike traditional methods that involve multiple steps—such as region proposals, feature extraction, and classification, this method processes the entire image in a single pass through a CNN. This way it is possible to predict simultaneously bounding boxes and class probabilities. The input image is divided into a grid, with each grid cell responsible for predicting objects whose centers fall within the cell. Each cell predicts:
 - Bounding box coordinates;
 - A confidence score indicating the likelihood of an object being present and the accuracy of the bounding box;
 - Class probabilities for object classification;

Unlike region-based methods like Fast R-CNN, YOLO considers the entire image during training. This allows a better understanding of the relationships between objects and their surroundings. To manage overlapping detections redundant bounding boxes are eliminated using non-maximum suppression (NMS) to ensure that only the most confident predictions are retained [21].

- **Single Shot MultiBox Detector (SSD):** SSD is a framework that performs object detection in a single forward pass through a CNN. It introduces default bounding boxes at multiple

scales and aspect ratios for each cell in the feature map. For each default box, SSD predicts class scores for object categories and adjusts the bounding box coordinates for accurate localization. To enhance detection across various object sizes, SSD adds extra convolutional layers to the base network, generating feature maps at multiple resolutions. High-resolution feature maps are used to detect smaller objects, while lower-resolution maps are better suited for larger objects. This multi-scale design makes SSD more effective at handling objects of different sizes compared to single-scale methods like YOLO. SSD also applies NMS to remove overlapping bounding boxes and retain only the most confident detections [22].

3.2 2D Image Segmentation Methodologies

Image segmentation is a fundamental problem in computer vision. The goal is to divide an image into distinct and coherent regions, or segments, which are meaningful with respect to a given task. Partitioning one image into segments facilitates higher-level analysis, such as object recognition, tracking, or scene understanding. However, defining what constitutes a “meaningful” segment is inherently ambiguous and depends heavily on context.

When humans perform image segmentation, the process is guided not only by low-level visual cues but also by perceptual organization principles. In 2008, Prof. Dejan Todorovic, presented a list of principles that mimic the human perception for image segmentation. These principles include [23]:

- **proximity:** grouping of spatially close elements.
- **similarity:** grouping of elements with similar attributes.
- **common fate:** grouping of elements that exhibit similar motion.
- **continuity:** preference for continuous curves and edges. If oriented units or groups are in alignment with one another, they are more likely to be incorporated into perceptual wholes.
- **connectivity:** grouping of elements within a bounded area or connected structure.
- **symmetry:** grouping based on symmetrical arrangements.

Human segmentation is inherently subjective, influenced by prior knowledge, context, and perceptual biases. These cognitive heuristics make human segmentation difficult to an algorithm. Translating these perceptual intuitions into computational procedures remains a central challenge in the development of robust and general-purpose segmentation algorithms.

3.2.1 Bottom-Up Segmentation

Bottom-up segmentation algorithms in 2D image segmentation are based on the principle that pixels are grouped into regions based on local similarities in appearance, such as colour, intensity, or texture. These methods start from the pixel level and progressively group similar elements into larger segments.

K-Means Clustering

K-Means is an unsupervised clustering algorithm that can be applied to 2D image segmentation by grouping pixels based on feature similarity. It divides the pixels into K non-overlapping clusters. Each pixel is assigned to the cluster with the nearest mean in the feature space [24].

- **Features Representation:** Each pixel is represented as a vector in a predefined feature space. The most common representations include colour features (e.g., RGB or Lab), and optionally spatial coordinates to promote spatial coherence in segmentation (e.g., [R, G, B, x, y]).
- **Clustering Process:** K-Means initializes K cluster centroids and iteratively assigns each pixel to the closest centroid based on Euclidean distance. Centroids are updated by calculating the mean of the assigned points. The process continues until convergence is achieved and the centroids positions in the feature map stabilizes.
- **Segmentation Output:** The final segmentation is obtained by assigning each pixel to its respective cluster label. The segmented image is produced by replacing each pixel with the mean value or representative label of its cluster.
- **Characteristics and Limitations:** The K-Means algorithm requires a predefined specified value of K , corresponding to the number of clusters. This implies that its success depends on prior knowledge of the expected number of segments in the feature space. Additionally, the algorithm's performance is influenced by both the initialization of the centroids and the choice of feature space. K-Means assumes that clusters are isotropic, convex, and separable in the feature space. This implies that the resulting clusters are approximately spherical, with point density decreasing from the centre toward the periphery, regardless of the direction of analysis. Such assumptions may not hold in the context of complex images.

Mean-Shift Segmentation

Mean-Shift is a non-parametric clustering algorithm used for 2D image segmentation. It identifies clusters by locating the modes (local maxima) of a density function in the feature space, without requiring prior specification of the number of clusters [25].

- **Features Representation:** Each pixel is mapped to a feature vector, commonly defined by colour (e.g., RGB or Lab) and optionally spatial information (e.g., [R, G, B, x, y]) to incorporate spatial continuity in the segmentation.
- **Clustering Process:** The algorithm defines a kernel window (usually Gaussian) and iteratively shifts each data point towards the mean of data points within its neighbourhood. This process continues until convergence and data points accumulate around the modes in the feature space.

- **Segmentation Output:** Pixels converging to the same mode are assigned the same label, forming coherent segments.
- **Characteristics and Limitations:** Mean-Shift does not require prior knowledge of the number of clusters, which distinguishes it from K-Means. It can identify an arbitrary number of clusters and is robust to outliers in the feature space. However, the algorithm is simple but computationally expensive.

The segmentation outcome is sensitive to the choice of the bandwidth parameter, which defines the kernel size. A small bandwidth may lead to over-segmentation, whereas a large bandwidth may merge distinct regions. Additionally, the algorithm assumes that clusters correspond to modes of a smooth density function. This assumption may not hold in the presence of abrupt changes or noise in the image.

A weighted mean can be used instead of a standard mean to reduce the influence of points near the outskirts of the kernel. This approach leads to smoother convergence, particularly when using higher-order kernels, compared to using the unweighted mean.

Felzenszwalb-Huttenlocher Graph-Based Segmentation

This method represents the image as an undirected graph and performs segmentation by iteratively merging regions based on an adaptive pairwise region comparison predicate. It achieves near-linear time complexity by combining local appearance cues with global consistency constraints [26].

- **Features Representation:** The image is modelled as a graph $G = (V, E)$, where each pixel is a node $v \in V$, and edges $(v_i, v_j) \in E$ connect adjacent pixels. Each edge is weighted by a dissimilarity measure such as the absolute difference in colour or intensity between connected pixels.
- **Segmentation Process:** The algorithm follows a bottom-up methodology, where regions are iteratively merged based on internal difference measurements. Initially, each pixel is treated as its own region of interest, meaning that every pixel starts as a separate component (segment).

Edges in the image graph are sorted in non-decreasing order by weight, allowing the algorithm to process the most similar (i.e., lowest-weight) pixel pairs first. This ordering helps preserve fine structural details before larger-scale groupings are considered.

At each step, the algorithm examines an edge (v_i, v_j) and considers merging the components C_i and C_j to which the vertices v_i and v_j belong. A merge occurs if the edge weight satisfies the condition:

$$w(v_i, v_j) \leq \min(\text{Int}(C_i) + \tau(C_i), \text{Int}(C_j) + \tau(C_j)) \quad (3.1)$$

In this formulation, $\text{Int}(C)$ denotes the maximum edge weight in the minimum spanning tree (MST) of component C , and $\tau(C) = \frac{k}{|C|}$ is a scale-dependent threshold determined by the component size and a global parameter k . The merge criterion in Equation 3.1 ensures that two regions are combined only when the connecting edge is not significantly stronger than the internal variability of either region.

- **Segmentation Output:** The algorithm returns a partition of the image where each segment corresponds to a connected component in the final graph. The result suppresses false segmentation in high-variability regions and maintains details in low-variability regions.
- **Characteristics and Limitations:** The algorithm does not require the number of segments to be predefined. The parameter k controls the scale of observation: larger values of k encourage larger regions. The algorithm is computationally efficient and operates in near-linear time with respect to the number of pixels. However, the segmentation quality is sensitive to the choice of k and assumes that segments can be adequately represented by low internal variation. The method uses the minimum edge weight to evaluate the difference between regions, which may be sensitive to single weak connections. Noise in the image can also affect the boundary accuracy.

Watershed Segmentation

Watershed segmentation interprets the image as a three-dimensional surface, where pixel intensities correspond to elevations. Regions are grown from initial local minima through a flooding simulation model, producing catchment basins that are separated by watershed lines. This separation prevents the merging of waters originating from different sources [27].

- **Features Representation:** A greyscale image is treated as a topographic surface, where each pixel's intensity corresponds to its elevation. Regional minima are used as the initial seeds for the flooding process, and the image domain is partitioned into catchment basins associated with these minima.
- **Segmentation Process:** The algorithm simulates a flooding process on the topographic surface. Water is allowed to rise from each minimum, and regions grow as water fills the basins. Pixels are processed in increasing order of intensity. A pixel is assigned to a basin if it is adjacent to a labelled pixel and has not yet been assigned a label. If it is adjacent to multiple basins with different labels, it is marked as a watershed pixel.

The procedure involves:

1. *Sorting Step:* All pixels are sorted by grey level using a bucket sort, providing efficient access during the flooding step.
2. *Flooding Step:* Beginning from each labelled minimum, a first-in-first-out (FIFO) queue propagates labels to neighbouring pixels, simulating the expansion of water

within catchment basins. Watershed pixels are marked when two or more flooding fronts meet.

- **Segmentation Output:** The output consists of labelled regions (basins) and watershed lines, which represent the boundaries between adjacent catchment basins. Depending on the application, watershed lines can either be retained or reassigned to the nearest region to produce a fully partitioned image.
- **Characteristics and Limitations:** Watershed segmentation is formally a region growing algorithm, as it:
 - *Seed Selection:* Uses regional minima or externally provided markers as initial seeds.
 - *Growth Criterion:* Expands regions based on increasing intensity levels, propagating labels to unassigned neighbouring pixels.
 - *Boundary Criterion:* Boundaries form where growth fronts from different regions meet.
 - *Boundary Handling:* Pixels at these interfaces are marked as watershed lines.

However, compared to classical region growing algorithms, which typically use fixed intensity difference thresholds or homogeneity criteria, watershed segmentation differs in that it is driven by the global topography of the intensity surface, not local similarity alone. It inherently builds a complete tessellation of the image domain.

A known limitation is the high sensitivity to noise and small local minima, which can result in over-segmentation. Marker-controlled watershed segmentation mitigates this by introducing user-defined or algorithmically derived markers to guide the flooding and suppress spurious regions.

The algorithm is computationally efficient, with a complexity linear in the number of pixels, and is suitable for various grid structures and higher-dimensional data [28], [29].

3.2.2 Top-Down Segmentation

Top-down segmentation algorithms are informed by higher-level knowledge about the scene or objects present in an image. Unlike bottom-up methods, which rely solely on local appearance cues, top-down segmentation uses semantic, contextual, or model-based information to guide the grouping of pixels into meaningful regions corresponding to known objects or parts.

Active Shape and Appearance Models (ASM / AAM)

Active Shape Models (ASM) and Active Appearance Models (AAM) are parametric statistical models that use a training set of annotated examples to capture variability in shape and appearance [30], [31].

- **Prior Knowledge Representation:** The shape of the object is typically represented as a set of landmark points, which are aligned and analysed using Principal Component Analysis (PCA) to build a linear shape model. ASM focuses solely on shape, while AAM also integrates texture information, constructing a combined model that describes how both shape and appearance vary together.
- **Segmentation Strategy:** The model is fit to a new image by iteratively adjusting shape and appearance parameters to minimize the difference between the predicted and observed image content. Optimization is typically gradient-based and sensitive to initialization.
- **Segmentation Output:** A parametrized contour or region that conforms to statistically plausible shapes and appearances, aligned to the image content.
- **Characteristics and Limitations:** Models encode strong shape constraints and enforce global consistency, which is advantageous in structured domains like facial or organ segmentation. However, the models accuracy is limited by the representational power of the training set. Poor generalization occurs if test shapes differ significantly from the training examples. Furthermore, annotating training images is labour-intensive.

Shape-Prior-Constrained Active Contours

Active contours (or snakes) are energy-minimizing curves that evolve under the influence of internal smoothness constraints and external image forces. In the standard form, the snakes are susceptible to weak edges and noise. To address this, top-down variants introduce shape priors to regularize the contour evolution, ensuring that the evolving boundary remains consistent with expected object shapes [32].

- **Prior Knowledge Representation:** Shape priors are incorporated by penalizing deviations from a reference shape, typically modelled through a set of training shapes and represented using statistical shape models (e.g., PCA). The active contour must simultaneously align with image features and remain close to the shape manifold defined by the training set.
- **Segmentation Strategy:** The active contour must simultaneously align with image features and remain close to the shape manifold defined by the training set.
- **Segmentation Output:** A closed contour approximating the object boundary while remaining within the learned shape constraints. The segmentation output is a smooth, closed contour that adheres to the object boundaries in the image while conforming to learned shape constraints. This method improves robustness in scenarios with poor contrast or missing boundary information.
- **Characteristics and Limitations:** This method is robust to scenes with missing edges and noise. Suitable for objects with limited intra-class variability and clearly defined shape priors. Although, the algorithm is sensitive to initialization and inflexible if the object true shape lies outside the learned shape distribution [33].

Model-Guided Deep Learning

Top-down deep learning-based segmentation methods, such as Mask R-CNN and U-Net, integrate object detection and instance segmentation by leveraging hierarchical features learned from large datasets [34], [35].

- **Prior Knowledge Representation:** The model incorporates prior semantic information through supervised learning. During training the model learns object shapes, appearances, and contextual relations. Unlike purely bottom-up deep networks, these top-down methods impose a hierarchical structure and semantic labelling process that mirrors object-centric segmentation tasks.
- **Segmentation Strategy:** In Mask R-CNN, the process begins with a Region Proposal Network (RPN) that identifies potential object locations. Each region is then classified, and a binary mask is predicted for each identified object instance. U-Net uses an encoder-decoder architecture with skip connections to combine high-level contextual information with low-level spatial detail, enabling accurate and dense pixel-wise predictions.
- **Segmentation Output:** The segmentation output consists of either class-specific semantic masks (U-Net) or instance-aware masks with bounding box localization (Mask R-CNN).
- **Characteristics and Limitations:** The models maintain a strong performance in complex visual scenes, effectively handling multiple object classes and occlusions. However, they rely on large annotated datasets and exhibit high computational costs during both training and inference. The ability to generalize across domains is limited. Performance typically degrades in scenarios with insufficient or significantly different data distributions unless fine-tuning or retraining is performed. These approaches are most applicable when detailed annotations are available and object-level semantics are critical to the task.

Chapter 4

Deep Learning Approach for Automatic Cullet Quality Estimation

This chapter presents the methodology developed for the automatic estimation of cullet sample quality using deep learning and image-based processing techniques. The proposed system is designed to estimate the class-wise material composition of a cullet sample by processing a single RGB image, without requiring any additional instrumentation or manual input. The architecture integrates three primary components: object detection, single-object classification, and region-based segmentation. Together, these modules form a unified pipeline capable of identifying individual particles, assigning material classes, and estimating the weight distribution of the sample based on visual data alone. The resulting outputs support downstream assessment of compliance with industrial quality control specifications.

The chapter is organized as follows. Section 4.1 describes the dataset acquisition protocol, including the experimental setup used to replicate industrial imaging conditions, the procedures for image rectification and annotation, and the data augmentation strategies employed to improve model generalization. Section 4.2 introduces the complete cullet quality estimation pipeline, beginning with the object detection module, followed by the classification model responsible for material identification, and concluding with the segmentation-based area estimation strategy and sample weight distribution estimation.

4.1 Dataset Creation

The performance of the deep learning architecture in real-world applications depends on the degree of similarity between the experimental data and actual industrial conditions. Owing to the complexity of the operational environment in industrial cullet treatment facilities, it was not feasible to collect data directly using the equipment employed in routine manual quality assessments. This procedure is described in Section 2.2.3, subsection *Quality Control Analysis for Cullet*. As an alternative, an experimental setup was developed to reproduce the visual and procedural characteristics of the industrial context. The setup comprised a camera mounted above a neutral white

tabletop. Data acquisition was performed using a Canon PowerShot SX420 IS camera. Figure 4.1 shows the experimental setup used.

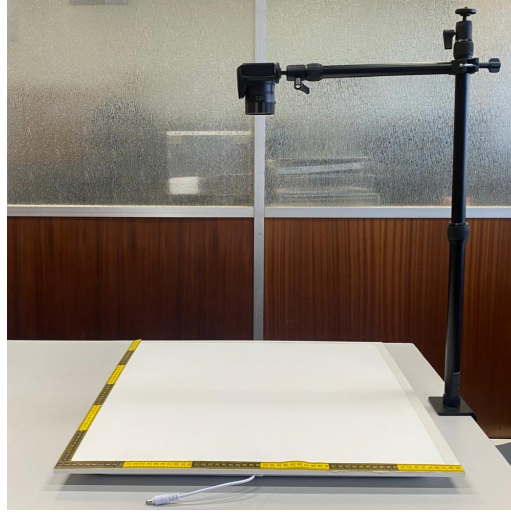


Figure 4.1: Experimental setup used for image acquisition.

The development and evaluation of the deep learning architecture relied on a dataset specifically created for the task. For each dataset instance, a cullet sample was collected and the particles were randomly distributed over the surface. The particles were manually rearranged in a single layer to avoid overlapping, and an image was captured. Following image acquisition, the entire sample was weighed, and the constituent particles were manually grouped by class and weighed again. The visual data was added to the dataset, while the associated weight distribution were recorded in a structured database. A unique sample identifier (ID) was assigned to each entry to ensure traceability between the image data and the corresponding weight distribution records.

During this work 92 images were collected using the experimental setup. The dataset was split as follows:

- **Training Set:** 73 images (80% of total dataset)
- **Validation Set:** 9 images (10% of total dataset)
- **Test Set:** 10 images (10% of total dataset)

Dataset Rectification

The camera used for image acquisition introduces optical distortion due to the internal configuration of its lens-sensor system. The degree and nature of this distortion depend on the specific characteristics of the camera's design. The distortion effect on the image data is the geometric deformation of the captured scene. Such distortion causes straight lines in the physical scene to appear curved in the captured image.

The objective of image rectification is to identify and correct this distortion. Rectification involves estimating the distortion parameters specific to the camera and applying a geometric

correction to the raw image data. This process compensates for the imperfect projection of the scene onto the image sensor caused by lens aberrations. The resulting rectified images provide a more accurate geometric representation of the physical scene.

The calibration process used to estimate the distortion was based on the method proposed by Zhang [36], implemented using the OpenCV library [37]. This method relies on calibration dataset constituted by several images of a known calibration target, a planar chequerboard with 9×6 inner corners and square size of 22mm, positioned at different orientations. The method estimates:

- Intrinsic Camera Parameters: describe the internal geometry of the camera (focal lengths, principal point);
- Extrinsic Parameters: describe the camera's position and orientation relative to the calibration target;
- Distortion Coefficients: model radial and tangential lens distortion effects.

The result of the calibration is the matrix-based camera model that describes how the camera maps 3D points in the scene onto the 2D image plane, including the distortion introduced by the lens.

In addition to the camera model, the calibration procedure also determines a millimetres-per-pixel (mm/pixel) conversion factor, which enables the translation of image-space measurements into real-world physical dimensions. This factor is calculated using the known physical size of the calibration chequerboard squares (22mm) and the corresponding distance between detected corner points in the image plane (in pixels). By averaging the pixel distances between adjacent corners along both horizontal and vertical axes across all calibration images and comparing them to the known square size, the mm/pixel ratio is estimated. This scale factor facilitates metric measurements in rectified images, supporting physically grounded analysis such as object size estimation.

To apply the model and rectify the dataset, an updated version of the intrinsic camera matrix was first computed to preserve image content while minimizing boundary artifacts. Based on the estimated distortion parameters, a geometric mapping was then generated to remap pixels to rectified positions. Finally, each image was processed using this mapping to produce rectified outputs free of lens distortion.

Calibration accuracy was evaluated using the mean re-projection error, which quantifies the discrepancy between the observed image points and the model-projected points. A low error value indicates that the estimated parameters accurately model the camera's projection behaviour.

This rectification process ensures that the geometric integrity of the dataset images is preserved, enabling more reliable image segmentation and analysis in subsequent stages.

Dataset Annotation

The dataset was annotated using the *Roboflow* framework, a computer vision platform that supports dataset preparation, annotation, and export for object detection, classification, and segmentation tasks. In this study, Roboflow was used to annotate each image with bounding boxes and associated class identifiers. The annotations were exported in COCO format [38], with each image linked to a corresponding text file containing the object labels and coordinates.

During the annotation process, a total of ten object classes were defined, each representing a material type found in cullet samples. The mapping between the original class identifiers and a rectified set of IDs was introduced to provide a more intuitive and consistent class order, minimizing the potential for confusion during model training and evaluation.

Class Name	Original ID	Rectified ID	Object Count
White Glass	9	1	1,728
Blue Glass	2	2	1,180
Other Whites Glass	7	3	137
Green Glass	3	4	284
Amber Glass	1	5	363
SCP	8	6	283
Opals	5	7	91
Plastics	6	8	17
Metals	4	9	150

Table 4.1: Mapping of original to rectified class IDs.

As shown in Table 4.1, the dataset is unbalanced in terms of class representation. The *White Glass* class, corresponding to transparent recycled glass fragments, is the most frequently observed, while the *Plastics* class is the most under-represented. This imbalance reflects the actual composition of flint cullet samples. As previously shown in Section 2.2.3 (Table 2.1), the majority of the material in the cullet is transparent glass, with contaminants appearing in smaller quantities.

To mitigate this imbalance, additional fabricated samples were created with an intentionally higher concentration of contaminants. These were included in the dataset to increase the number of examples from under-represented classes. Despite this intervention, the dataset remains imbalanced. Consequently, it is expected that the deep learning model may demonstrate reduced classification performance for under-represented classes and a tendency to misclassifying them as instances of overrepresented categories.

Data Augmentation

To account for the variability observed in industrial data acquisition conditions and increase the generalizability of the trained model, a set of data augmentation techniques was applied during

training. These transformations modify the input image data in controlled ways, simulating environmental and operational variability that may be encountered in real deployment scenarios. The augmentations used are described below:

- **Geometric Transformations:** Horizontal and vertical flipping operations were applied to both the image data and the associated object bounding boxes. These transformations reflect the spatial arrangement of cullet particles across the respective axes. In the context of cullet analysis, where particle orientation is arbitrary and non-informative, this augmentation reduces sensitivity to orientation-dependent features and counteracts any positional patterns unintentionally introduced during manual sample preparation or imaging.
- **Mosaic Augmentation:** Mosaic augmentation combines four distinct training images into a single composite image, along with their respective bounding box annotations. This technique increases the spatial diversity within a single training sample and creates scenarios with a higher density of objects. Given that the cullet sample weight is approximately proportional to the number of individual particles per image, this augmentation simulates conditions of higher particle density. Such conditions may result either from larger samples per analysis or finer particle granulometry, both relevant variables in industrial applications.
- **Illumination Variation:** Controlled adjustments were applied to image brightness and contrast parameters to simulate variation in lighting conditions. This reflects the inherent illumination variability in industrial environments, where lighting conditions are influenced by factors such as ambient light fluctuations, reflective surfaces, and changes in the imaging setup. This augmentation reduces model sensitivity to illumination-specific cues present in the training data.
- **Motion Blur:** Motion blur was synthetically applied by convolving the image with a directional kernel to simulate motion artifacts. This transformation accounts for blur effects that may occur due to relative motion between the imaging system and the cullet particles. Such conditions may arise in scenarios where the system is integrated directly into a conveyor-based production line, in which camera vibration or particle displacement can introduce blur. This augmentation prepares the model to handle these motion-induced artifacts.

Given the limited original dataset size (92 images), the application of these augmentations is intended to enhance robustness without misleading the model with unrealistic distortions. As augmentation introduces diversity without introducing new semantic content, the dataset was expanded by a factor of two. This scale was chosen to increase representativeness while preserving the relative distribution of classes and avoiding oversampling. Further augmentation beyond this threshold was not pursued to minimize the risk of learning biases from repeated synthetic patterns. This approach aims to strike a balance between generalization capability and data integrity.

4.2 Automatic Cullet Quality Estimation

This section describes the pipeline developed for automatic cullet quality estimation using image-based analysis. The system integrates object detection, classification, and segmentation components to estimate the material composition of cullet samples and assess their compliance with quality standards. An overview of the entire system architecture is presented in Figure 4.2.

The proposed methodology is designed to fulfil two primary objectives, the detection and classification of individual particles in an image, and the estimation of sample quality based on the class-wise distribution of mass. The pipeline receives a digital image of a cullet sample as input and performs the following operations:

- **Object Detection:** The *Multi-Object Bounding Box Proposal* module employs a convolutional neural network to identify all visible particles in the scene, generating bounding boxes that define the spatial extent of each object.
- **Object Classification:** Each detected particle is classified into one of the predefined material categories using a dedicated *Single-Object Classification* model. This step associates each object with a material label, distinguishing between recyclable glass types and contaminants.
- **Segmentation and Area Estimation:** A Chan–Vese segmentation algorithm is applied to each particle in order to estimate its area in pixels. The segmented areas are grouped by class to obtain the total segmented area per material category.
- **Mass Estimation:** The aggregated areas are multiplied by class-specific particle density values, which are precomputed from the training dataset. This yields an estimate of the total mass associated with each class in the sample image.
- **Quality Assessment:** The resulting class-wise weight distribution is used to evaluate the overall sample quality according to the predefined acceptance criteria outlined in Section 2.2.3.

Each component operates independently but contributes to the cumulative estimation of material composition. The methodology is designed to function entirely from visual data and requires no additional physical measurements beyond the input image. The following subsections describe each module of the pipeline in detail, along with its implementation, parameters, and functional role in the system.

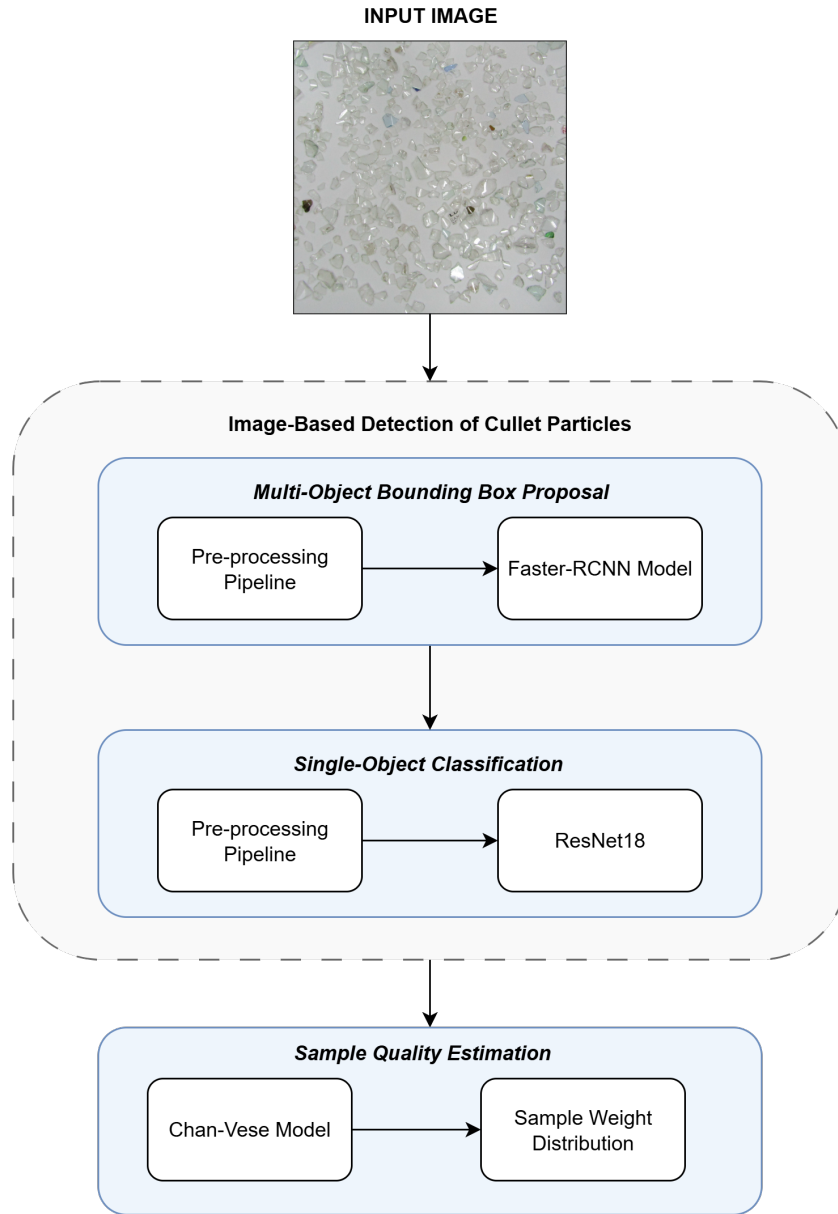


Figure 4.2: Overview of the proposed image-based cullet quality estimation pipeline. The system integrates object detection, classification, and segmentation modules to compute the class-wise weight distribution for each sample image.

4.2.1 Multi-Object Bounding Box Proposal

The multi-object bounding box proposal pipeline aims to automatically detect all visible particles within an input image of a cullet sample. This stage constitutes the first step in the automatic cullet quality estimation process. Given a single input image, the system processes the image and determines a set of bounding boxes corresponding to the individual cullet particles, regardless of their class. The bounding boxes serve as the foundation for subsequent tasks, including classification

and weight estimation. This component of the pipeline is implemented using a deep learning-based object detection architecture trained to identify all discrete material fragments present in each sample image. The pipeline detailed description is presented in this section.

Pre-Processing Pipeline

Prior to model training and inference, all input images underwent a systematic pre-processing pipeline designed to enhance the quality and consistency of the visual data. This pipeline aims to enhance local features relevant to object detection, standardize input dimensions, and decrease computing complexity. The entire dataset was subjected to the following consistent procedures:

- **Image Resizing:** The original images used in this study have a resolution of 5153x3864 pixels, corresponding approximately to a 4:3 aspect ratio. To preserve this geometric proportion while optimizing computational efficiency, a resizing protocol was applied in which the width was fixed to 800 pixels. The length was scaled proportionally to approximately 1067 pixels (calculated as $\frac{4}{3} \times 800$) in order to maintain the original aspect ratio.

This approach ensures that the spatial structure of image content is preserved, which is essential for the accurate localization and classification of cullet particles. The width constraint of 800 pixels was selected as the limiting dimension to reduce computational load and memory usage during training. By reducing the image dimensions, this method maintains sufficient detail for detecting small and medium-sized particles while enabling feasible training times on the available hardware [18].

- **Contrast Limited Adaptive Histogram Equalization:** Contrast Limited Adaptive Histogram Equalization (CLAHE) is a technique used to enhance local contrast in digital images. The method divides the image into small, non-overlapping rectangular regions, referred to as tiles. Histogram equalization is then applied independently to each tile, allowing for localized contrast adjustments. To avoid discontinuities at tile boundaries, bilinear interpolation is used to blend adjacent regions, reducing edge artifacts and promoting smooth transitions across the image [39].



(a) Image without CLAHE



(b) Image with CLAHE

Figure 4.3: Comparison of images before and after CLAHE.

As shown in Figure 4.3, CLAHE improves the contrast between the cullet and the background, with a particularly noticeable effect on transparent glass fragments. This enhancement increases both the number and the complexity of image features, such as edges, gradients, and fine texture variations, which strengthens the input to the deep learning feature extraction pipeline. The increase in availability of distinct local features contributes to more accurate class characterization and improves discrimination between classes. Additionally, by sharpening object boundaries, CLAHE facilitates the segmentation task, as well-defined contours improve the delineation of individual particles. However, the method also amplifies shadows caused by uneven illumination in the experimental setup and shifts the colour balance. As a result, some materials, such as white ceramics, may appear more visually similar to transparent white glass, potentially introducing ambiguity in classification, potentially complicating the classification task in some scenarios.

Object Detection Model

A Faster R-CNN architecture was adopted for the object detection task. The model is composed of a ResNet-50 backbone with a Feature Pyramid Network (FPN), initialized with pre-trained weights from the COCO dataset [40]. To adapt the model to the specific requirements of this task, the classification head was modified to perform binary classification, distinguishing between a single target class (cullet) and background. Following initialization, the model was fine-tuned on the custom dataset developed for this work, allowing the network to adjust its parameters to the visual and structural characteristics specific to cullet samples. An overview of the training configuration and parameters choices is presented in Table 4.2.

Training was conducted using a batch size of one. This configuration was selected due to the limited size of the dataset (92 images). The use of a batch size of one has been shown to improve generalization in small-data regimes by increasing gradient variance and reducing the likelihood of over-fitting [41], [42].

Despite the dataset being augmented to increase its diversity and simulate realistic operational variability, training was conducted using a batch size of one. This decision was primarily motivated by the limited size of the original dataset (92 annotated images). Using a batch size of one introduces higher variance in the gradient updates, which can act as a form of implicit regularization and has been shown to improve generalization in small-data regimes by preventing early convergence to sharp local minima [41], [42]. However, this configuration also introduces limitations, higher gradient variance can slow convergence, and the lack of batch-level statistics precludes the use of normalization strategies such as Batch Normalization. Although data augmentation increases sample diversity, maintaining a batch size of one ensures that each augmented image contributes independently to the gradient computation. This approach helps preserve gradient variability and avoids introducing artificial correlations that may arise when multiple augmented samples are processed simultaneously in a single batch.

Stochastic Gradient Descent (SGD) was used as the optimization algorithm. A learning rate of 0.005 was defined to balance training stability and convergence rate. Momentum was set to

0.9 to smooth the optimization path and accelerate convergence by incorporating the influence of previous gradients. L2 regularization was introduced through a weight decay coefficient of 0.0005 to reduce over-fitting by penalizing large parameter values.

The learning rate was scheduled to decay by a factor of 10 every three epochs using a step-based scheduler. This adjustment helps stabilize training and refine the model's convergence in later epochs.

Training proceeded for a maximum of 25 epochs. Each epoch included a training phase and a validation phase. An early stopping mechanism was implemented to monitor validation set performance. If the validation Intersection over Union (IoU) did not improve by at least 0.01 for five consecutive epochs, training was terminated. The model state with the best validation IoU was saved.

During inference, a global threshold of 0.9 was applied to the detection scores. Only predictions above this confidence level were retained.

All relevant training metrics, including loss values and IoU scores for both training and validation, were recorded throughout the process. The final trained model was saved for further evaluation and deployment.

Parameter	Value
Architecture	Faster R-CNN (ResNet-50 + FPN)
Pretrained Weights	COCO (COCO_V1)
Batch Size	1
Optimizer	SGD
Learning Rate	0.005
Momentum	0.9
Weight Decay	0.0005
LR Scheduler	StepLR (step size = 3, gamma = 0.1)
Epochs	25
Early Stopping Patience	5 epochs (min Δ IoU = 0.01)
Detection Threshold	0.9

Table 4.2: Summary of training parameters and configurations.

Although the dataset created in this work is fully annotated for multi-class object detection, the object detection model was trained and deployed in a single-class (binary) detection setting. This choice was motivated by empirical observations during model development and testing. Initial experiments using the full multi-class labelling scheme revealed a substantial decline in the model ability to detect cullet particles altogether. In the multi-class context, the model failed to generate bounding boxes for a portion of the particles, independent of classification accuracy. The class imbalance in the dataset further exacerbated this problem, limiting the model ability to learn meaningful inter-class distinctions.

These observations are consistent with findings in the literature, where object detection models trained on small and imbalanced datasets exhibit degraded performance in localization, especially for under-represented classes or when discriminative features are subtle or highly correlated across classes [43], [44]. When the model attempts to differentiate multiple classes in such conditions, the shared capacity of the backbone and classifier heads becomes insufficient to learn both accurate localization and reliable fine-grained classification. Consequently, simplifying the task to binary detection allowed the model to focus exclusively on detecting all particles, regardless of class.

This adjustment aligns with the primary objective of the detection stage, to identify all visible cullet particles in the scene. The fine-grained classification is deferred to subsequent stages.

4.2.2 Single-Object Classification

The classification stage assigns a class to each cullet particle identified in a sample image. It operates on the output of the object detection model, which provides bounding boxes delineating the location of each visible object. For each bounding box, the corresponding image region is extracted and processed independently by the classifier. This process enables the systematic categorization of all detected objects in the cullet sample.

Pre-processing Pipeline: Cropping Input Image for Single-Object Classification

The ResNet-18 architecture used for the classification stage is designed for single-object image classification, where each input is assumed to contain one dominant object. To comply with this constraint, a dedicated pre-processing pipeline was implemented to convert multi-object images into isolated single-object instances.

During inference, this procedure involves cropping each original image using the bounding boxes predicted by the object detection model. For every particle detected, the bounding box coordinates define a region of interest, which is extracted from the original image. A fixed padding margin is applied around each bounding box to preserve limited contextual information. As a result, each input image containing multiple objects is transformed into a set of cropped images, each ideally containing a single, dominant object.

During training, however, the same cropping procedure is applied using ground truth annotations instead of predicted bounding boxes. This ensures that the bounding boxes used for cropping are optimal, precisely delineating each object. Each cropped image is assigned to its corresponding ground truth class label and the identifier of the source image, ensuring traceability to the full cullet sample. These annotations serve as supervised targets, enabling the computation of classification loss through comparison with the predicted labels. The resulting loss values guide the adjustment of model weights via backpropagation, progressively refining the model's ability to discriminate between material classes. This modification was introduced to decouple the performance of the classification model from the upstream object detection model during training. This allows the model to focus exclusively on learning class-discriminative features, independent of detection uncertainty. Eliminating any variance introduced by errors in object localization, such as

missed detections, misaligned bounding boxes, or inclusion of background or adjacent particles, that could otherwise harm the feature extraction process.

This pre-processing step is essential to align the dataset format with the architectural constraints of ResNet-18. Without isolating individual objects, the model would receive inputs containing multiple instances, violating the assumption of single-label supervision and introducing ambiguity into the classification process. Cropping ensures that each training and inference sample corresponds to a single object with a clearly defined label, supporting reliable multi-class classification.

Object Classification Model

A ResNet-18 (Residual Network) convolutional neural network was adopted for the single-object classification task. The standard ResNet-18 model is composed of 18 learnable layers, including one initial convolutional layer followed by four sequential stages, each consisting of two residual blocks. Each residual block comprises two convolutional layers with batch normalization and ReLU activation functions, interconnected by identity-based skip connections that facilitate gradient flow during training and improve optimization stability [45].

The original architecture concludes with a global average pooling layer followed by a fully connected (FC) classification layer designed for 1,000-class classification on the ImageNet dataset. In the context of this work, this final FC layer was replaced with a new fully connected layer consisting of nine output units, corresponding to the specific material categories identified in cullet classification, as listed in Table 4.3. This modification was required to adapt the network to the multi-class classification problem defined for this work context.

The model was initialized with weights pre-trained on the ImageNet dataset [46], which provides a broad feature representation due to its scale and diversity. To specialize the model for the domain-specific task, the pre-trained network was fine-tuned on the custom dataset. This step enabled the model to adjust its parameters to the visual and structural properties of cullet materials and to discriminate effectively between the nine target classes defined in this study.

Class ID	Class Name
0	White Glass
1	Blue Glass
2	Other White Glass
3	Green Glass
4	Amber Glass
5	Stones/Ceramics/Porcelain
6	Opals
7	Organic/Plastics
8	Metals

Table 4.3: Class label mapping used in the single-object classification task.

The model was fine-tuned using a supervised learning approach, an overview of the training configuration and parameters choices is presented in Table 4.4. The loss function used was Cross Entropy Loss, which is standard for multi-class classification problems. Training was conducted using Stochastic Gradient Descent (SGD) with a learning rate of 0.001, momentum set to 0.9, and L2 regularization via a weight decay coefficient of 0.0001 [47].

A *ReduceLROnPlateau* learning rate scheduler was employed, which reduced the learning rate by a factor of 0.5 when the validation macro F1-score did not improve for three consecutive epochs. This adaptive learning rate strategy supports convergence by allowing finer adjustments in later stages of training [48].

Training was run for a maximum of 100 epochs with early stopping based on the validation macro F1-score. If the validation score failed to improve over five consecutive epochs, training was interrupted. The model state corresponding to the highest validation F1-score was preserved.

Throughout training, performance metrics such as accuracy, loss, and macro F1-score were computed and logged for both training and validation sets. These metrics provide insight into the model classification capability across all classes.

Parameter	Value
Architecture	ResNet-18
Pretrained Weights	ImageNet
Classification Task	Multi-class (9 classes)
Loss Function	Cross Entropy
Optimizer	SGD
Learning Rate	0.001
Momentum	0.9
Weight Decay	0.0001
LR Scheduler	ReduceLROnPlateau (factor=0.5, patience=3)
Epochs	100
Early Stopping Patience	5 epochs
Evaluation Metric	Macro F1-score

Table 4.4: Summary of training parameters and configurations for the multi-class classification model.

4.2.3 Sample Quality Estimation

This section presents the methodology used to estimate the composition of cullet samples using only image data. The goal is to approximate the class-wise weight distribution by combining detection, classification, and segmentation outputs. The object detection model from the *Multi-Object Bounding Box Proposal* step identifies particle locations, while the *Single-Object Classification* model assigns class labels. A segmentation algorithm is then applied to each localized particle to compute its area. These segmented areas are aggregated by class and combined with class-wise particle density values, precomputed from the training dataset, to estimate the total

mass per class within the sample. This completes the pipeline for automated quality evaluation of the cullet according to the criteria outlined in Section 2.2.3.

Chan-Vese Segmentation Model

The Chan-Vese model blurs the methodological distinction between top-down and bottom-up segmentation through its hybrid nature, combining elements of both paradigms. While inheriting aspects of top-down segmentation, especially through its model-driven framework, simultaneously employs data-driven region analysis that is characteristic of bottom-up approaches. The algorithm segments an image by evolving a contour that minimizes an energy functional based on regional intensity statistics. In this work, the evolution is performed using a level set formulation in which the contour is implicitly represented as the zero level set of a signed distance function [49].

This approach incorporates several top-down characteristics:

- **Initialization Requirement:** The level set function is initialized with a predefined binary region, corresponding to a centred rectangle. This reflects an assumption about the approximate object location, which directly influences the evolution trajectory and the final segmentation result.
- **Model-Based Energy Minimization:** The model segments the image by minimizing the energy function based on a piecewise-constant model, which assumes that the image consists of two homogeneous regions, each characterized by a constant intensity. This is a top-down assumption about image structure, not driven from the image data.
- **Contour Regularization:** The contour evolution is smoothed by a curvature-based regularization term, implemented by computing the divergence of the normalized gradient of the level set function. This term penalizes irregular or fragmented boundaries and promotes smooth, regular contours. Although it operates locally, this component reflects a top-down characteristic of the model, as it imposes geometric constraints on the shape of the segmentation rather than deriving them from image data.

Despite its top-down foundation, the Chan-Vese model incorporates key bottom-up features, making it partially data-driven:

- **Region-Based Intensity Adaptation:** Rather than relying on edge gradients (as in classical snakes), the Chan-Vese model evaluates pixel intensities over regions. At each iteration, the algorithm computes mean intensity values c_1 and c_2 for the interior and exterior of the evolving contour, respectively. These values are derived directly from the image data, allowing the model to adapt to local intensity variations instead of depending on preset templates or prior shapes.
- **Gradient-Free Evolution:** The segmentation is driven by the similarity of pixels to the regional means. The method does not rely on gradient information, making it more suitable than classical edge-based models, such as gradient-driven snakes, for images with low contrast or diffuse transitions.

In the context of this work, the ability to segment low-contrast particles, such as neutral or *white* glass, is particularly important. These particles often exhibit minimal intensity differences from the background, making edge-based methods unreliable for consistent segmentation.

- **Unsupervised Intensity-Based Adaptation:** The evolution of the contour is influenced by the discrepancy between the local pixel intensity and the average intensities of the regions defined by the contour. The term λ guides the contour toward separating regions where the intensity values are most dissimilar from the respective region averages.

$$\lambda = (I - c_1)^2 - (I - c_2)^2 \quad (4.1)$$

The term λ is calculated at each iteration of the algorithm from the equation 4.1. The variables c_1 and c_2 represent the mean intensities inside and outside the contour, respectively. Rather than explicitly assigning pixels to regions, this term acts as a continuous force: pixels with intensities closer to c_1 tend to pull the contour outward, while those closer to c_2 pull it inward. As a result, the model becomes sensitive to deviations from regional homogeneity and adapts the contour accordingly. This behaviour reflects a bottom-up characteristic of the algorithm since contour evolution is directly driven by the local distribution of image intensities rather than by external priors or predefined shapes.

Chan–Vese Algorithm Implementation

The algorithm implemented is a simplified variant of the original Chan–Vese model, excluding area-based terms and auxiliary forces such as balloon terms. The algorithm 1 outlines the segmentation procedure, which follows a region-based active contour approach. The method evolves an implicit contour by iteratively updating the level set function ϕ , driven by curvature regularization and intensity-based forces. The model receives an image as input, segments the object of interest and computes the object area in pixels. These segmentation results are used as input for the subsequent algorithm that estimates the object weight and determines the weight distribution for the sample.

The level set function is initialized as a binary function, equation 4.2, and the contour is defined as the zero level set of ϕ ($\phi(x, y) = 0$).

$$\phi(x, y) = \begin{cases} +1, & \text{if } (x, y) \in \text{centered box of size } b \times b \\ -1, & \text{otherwise} \end{cases} \quad (4.2)$$

The method iteratively updates ϕ to minimize the energy functional, balancing contour smoothness and data fidelity. The energy function, equation 4.3, maps the level set function (ϕ) to a scalar

value. Quantifying how well a given contour segments the image domain based on piecewise constant intensity regions.

$$E(c_1, c_2, \phi) = \mu \cdot \text{Length}(\partial\Omega) + \lambda_1 \int_{\text{inside}} |I(x, y) - c_1|^2 dx dy + \lambda_2 \int_{\text{outside}} |I(x, y) - c_2|^2 dx dy \quad (4.3)$$

The algorithm receives the following inputs:

- I : Input greyscale image.
- N : Number of iterations for level set evolution.
- b : Initial box size defining the starting contour region.
- μ, ν : Weighting parameters for smoothness and length regularization.
- ε : Regularization parameter for smoothed Dirac and Heaviside functions.
- Δt : Time step controlling the evolution speed.

Each iteration proceeds as follows:

- **Dirac delta approximation:**

$$\delta_\varepsilon(\phi) = \frac{\varepsilon}{\pi(\varepsilon^2 + \phi^2)} \quad (4.4)$$

This smooth approximation of the Dirac delta function localizes updates to a narrow band around the zero level set, ensuring numerical stability and computational efficiency by restricting modifications near the contour [50].

- **Heaviside function approximation:**

$$H_\varepsilon(\phi) = \frac{1}{2} \left(1 + \frac{2}{\pi} \arctan \left(\frac{\phi}{\varepsilon} \right) \right) \quad (4.5)$$

The Heaviside function differentiates the inside and outside regions for computation of region statistics, avoiding discontinuities inherent in the ideal Heaviside step function.

- **Gradient and curvature computation:** The spatial gradients of ϕ are computed as $\nabla\phi = (\phi_x, \phi_y)$, equation 4.6. The gradient is normalized with a small positive constant δ to avoid division by zero.

$$\mathbf{N} = \frac{\nabla\phi}{|\nabla\phi| + \delta} \quad (4.6)$$

The curvature of the level set, equation 4.2.3, is calculated as the divergence of the normal field.

$$\kappa = \nabla \cdot \mathbf{N} \quad (4.7)$$

- **Length regularization term:** The length regularization term, equation 4.2.3, in the Chan–Vese model serves to minimize the total length of the evolving contour, encouraging smooth, regularized boundaries during the segmentation process.

$$L = \nu \cdot \delta_\epsilon(\phi) \cdot \kappa \quad (4.8)$$

The parameter $\nu > 0$ is a weighting parameter that controls the influence of this term.

- **Penalty term for smoothness:**

$$P = \mu \cdot (\Delta\phi - \kappa) \quad (4.9)$$

This penalty term, equation 4.9, enforces that ϕ remains close to a signed distance function during evolution, improving numerical stability and reducing the need for reinitialization [50].

- **Mean intensity computations:** The mean intensities inside and outside the contour, c_1 and c_2 respectively, are calculated from the equation 4.10.

$$c_1 = \frac{\sum I \cdot H_\epsilon(\phi)}{\sum H_\epsilon(\phi)}, \quad c_2 = \frac{\sum I \cdot (1 - H_\epsilon(\phi))}{\sum (1 - H_\epsilon(\phi))} \quad (4.10)$$

- **Data fitting term:** The data fitting term, equation 4.11, guides the active contour evolution based on the difference between the image intensities and the mean intensities c_1 and c_2 .

$$F = \delta_\epsilon(\phi) \cdot [-(I - c_1)^2 + (I - c_2)^2] \quad (4.11)$$

When a pixel's intensity $I(x, y)$ is more closer to c_1 than c_2 , the fidelity parameter $\lambda = [-(I - c_1)^2 + (I - c_2)^2]$ becomes positive, pushing the zero level set of ϕ to include that pixel. Conversely, if the pixel intensity is closer to c_2 , the value becomes negative, pushing the contour outward to exclude that pixel from the segmented region.

- **Level set update:**

$$\phi \leftarrow \phi + \Delta t \cdot (L + P + F) \quad (4.12)$$

The level set function is updated by combining the smoothing, penalty, and data fitting terms, repeated for N iterations to progressively evolve the contour.

After completion of iterations, the binary mask is generated by thresholding ϕ , as showed in equation 4.13

$$M(x, y) = \begin{cases} 1, & \phi(x, y) \geq 0 \\ 0, & \text{otherwise} \end{cases} \quad (4.13)$$

Median filtering is applied to M to reduce noise. The external contours in M are extracted, with the largest contour selected as the object boundary. The area, in pixels, enclosed by this contour is then calculated.

Algorithm 1: Chan–Vese segmentation for object extraction

Input: Greyscale image I , number of iterations N , initial box size b , parameters μ, ν, ε , step size Δt

Output: Binary mask M , object contour C , object area A

Initialize level set function $\phi \leftarrow -1$ everywhere

Set $\phi = +1$ inside a centered box of size $b \times b$

for $i = 1$ to N **do**

 Compute Dirac delta approximation: $\delta_\varepsilon(\phi) \leftarrow \frac{\varepsilon}{\pi(\varepsilon^2 + \phi^2)}$

 Compute Heaviside approximation: $H_\varepsilon(\phi) \leftarrow \frac{1}{2} \left(1 + \frac{2}{\pi} \arctan \left(\frac{\phi}{\varepsilon} \right) \right)$

 Compute image gradients: $(\phi_x, \phi_y) \leftarrow \nabla \phi$

 Compute gradient magnitude $|\nabla \phi|$ and normal vectors (N_x, N_y)

 Compute curvature $\kappa \leftarrow \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi| + \delta} \right)$

 Compute length regularization term: $L \leftarrow \nu \cdot \delta_\varepsilon(\phi) \cdot \kappa$

 Compute penalty term for smoothness: $P \leftarrow \mu \cdot (\Delta \phi - \kappa)$

 Compute mean intensities: $c_1 \leftarrow \frac{\sum I \cdot H_\varepsilon(\phi)}{\sum H_\varepsilon(\phi)}$, $c_2 \leftarrow \frac{\sum I \cdot (1 - H_\varepsilon(\phi))}{\sum (1 - H_\varepsilon(\phi))}$

 Compute data fitting term: $F \leftarrow \delta_\varepsilon(\phi) \cdot [-(I - c_1)^2 + (I - c_2)^2]$

 Update level set: $\phi \leftarrow \phi + \Delta t \cdot (L + P + F)$

end

Create binary mask: $M(x, y) \leftarrow 1$ if $\phi(x, y) \geq 0$, else 0

Apply median filtering to M to reduce noise

Detect external contours in M

Select the largest contour C by area

Compute object area $A \leftarrow$ area of C

return M, C, A

4.2.4 Sample Weight Distribution Estimation

During the data acquisition, Section 4.1, for each cullet sample, the weight distribution per class was calculated. This information was stored in a structured database linked to image data by the image ID. This setup enables, for each image, the association of the number of pixels occupied by each class with the corresponding mass of that class in grams.

Using the automatic cullet quality estimation pipeline proposed in this work, it is possible to determine the class-wise sum of pixels per image. For an input image, the object detection model is used to localize each particle in the image, followed by classification to assign a class label. The Chan-Vese model is used to compute the individual area for each particle and the resulting areas are associated by class. Since the total mass per class for each sample is already known from the data acquisition the class-wise particle density can be calculated for this image using the equation 4.14.

$$\rho_{\text{pixels}}^{(c)} = \frac{m^{(c)}}{A_{\text{pixels}}^{(c)}} \quad (4.14)$$

The class-wise particle density ($\rho_{\text{pixels}}^{(c)}$) is define in g/pixels². By multiplying the $\rho_{\text{pixels}}^{(c)}$ by the square of conversion factor (conversion factor_{mm/pixel}), calculated during the dataset calibration, converts the class-wise particle density to a physical area unit (mm²), as defined in equation 4.15. This conversion factor, retrieved during the dataset rectification step, Section 4.1, defines the image scale in millimetres per pixel.

$$\rho_{\text{mm}^2}^{(c)} = \rho_{\text{pixels}}^{(c)} \cdot (\text{conversion factor}_{\text{mm/pixel}})^2 = \frac{m^{(c)}}{A_{\text{mm}^2}^{(c)}} \quad (4.15)$$

The next step is to apply the same rational to all the images in the training dataset. After determine the class-wise particle density for all the training data, is computed the mean class-wise particle density that is defined as the expected particles density for the all the cullet particles.

$$\bar{\rho}_{\text{mm}^2}^{(c)} = \frac{1}{N_c} \sum_{i=1}^{N_c} \rho_{\text{mm}^2,i}^{(c)} \quad (4.16)$$

The equation 4.16 computes the arithmetic mean of the per-image class-wise particle densities across all valid training samples. A sample is considered valid if it contains both measurable mass and a segmented area for the given class. The resulting value, $\bar{\rho}_{\text{mm}^2}^{(c)}$, represents the expected particle density for class c and is used as a class-specific parameter in the estimation of sample composition.

For a cullet sample with unknown composition, the corresponding image is processed using the complete segmentation and classification pipeline. The pipeline calculates the total segmented area of particles, grouped by class, from the input image. For each class c , the estimated mass is then computed by multiplying the total area (in pixels) by the expected particle density $\bar{\rho}_{\text{mm}^2}^{(c)}$. This procedure yields an estimate of the class-wise weight distribution within the sample.

Finally, the estimated weight distribution can be evaluated according to the quality control requirements outlined in Section 2.2.3, enabling a quantitative assessment of cullet sample quality.

Chapter 5

Experimental Results and Performance Evaluation

This chapter presents the experimental results obtained from the implementation of the proposed deep learning-based pipeline for automatic cullet quality estimation. The primary objective of this chapter is to evaluate the performance of each component within the pipeline and to assess their cumulative impact on overall system accuracy. Performance is measured using appropriate quantitative metrics.

5.1 Evaluation Metrics for Object Detection

To quantitatively evaluate the performance of the Faster R-CNN object detection model in the context of cullet particle localization, a set of standard metrics was adopted. These metrics assess the model's ability to detect and localize cullet objects in input images by comparing the predicted bounding boxes to manually annotated ground truth boxes. Evaluation is performed on a per-image basis, and results are averaged across the dataset. The following metrics were considered:

- **Average Intersection over Union (IoU@0.9):** The Intersection over Union (IoU) quantifies the spatial overlap between a predicted bounding box and a ground truth bounding box. It is defined as the ratio of the area of intersection to the area of union between the two boxes. detection is classified as correct if the IoU exceeds a fixed threshold. In this work, a threshold of 0.9 was used to enforce strict localization. The mean IoU across all correctly matched pairs provides an indication of average localization precision.

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}} \quad (5.1)$$

- **Average Precision at IoU 0.9 (AP@0.9):** Average Precision (AP) summarizes the trade-off between precision and recall at varying detection confidence thresholds. It is computed by integrating the precision-recall curve. For this task, AP is evaluated at a fixed IoU threshold of 0.9. This metric provides a threshold-agnostic measure of detection performance,

particularly useful for evaluating object presence in binary detection tasks such as cullet identification.

$$AP@0.9 = \int_0^1 \text{Precision}(r) dr \quad (5.2)$$

Detection Outcome Classification

Model predictions are categorized into four possible outcomes based on their agreement with ground truth annotations. These outcomes form the basis of a confusion matrix, as shown in Table 5.1.

Table 5.1: Interpretation of Detection Outcomes in Terms of the Confusion Matrix

Ground Truth	Predicted as Object	Predicted as Background
Object	True Positive (TP) $\text{IoU} \geq 0.9$	False Negative (FN) Missed detection
Background	False Positive (FP) $\text{IoU} < 0.9$ or spurious box	True Negative (TN) Correct rejection

A True Positive (TP) corresponds to a predicted bounding box that correctly localizes a cullet object with $\text{IoU} \geq 0.9$. A False Positive (FP) occurs when a predicted box does not match any ground truth object or overlaps insufficiently. A False Negative (FN) arises when a ground truth object is not detected by any prediction with sufficient IoU. A True Negative (TN) refers to background regions correctly left unmarked by the model. However, TNs are not explicitly counted in object detection due to the absence of exhaustive background annotations.

- **Accuracy:** Accuracy is the proportion of all correctly classified instances over the total number of instances. In object detection, accuracy is not always meaningful due to the typically large and undefined number of TNs. It is reported here for completeness but should be interpreted with caution in this application.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \quad (5.3)$$

- **Precision:** Precision quantifies the proportion of correctly predicted cullet particles among all predicted objects. High precision indicates a low rate of false positives and suggests reliable detection of actual cullet instances.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (5.4)$$

- **Recall:** Recall measures the proportion of ground truth cullet particles that were successfully detected. High recall signifies that most true objects were detected by the model.

$$\text{Recall} = \frac{TP}{TP + FN} \quad (5.5)$$

- **F1 Score:** The F1 score is the harmonic mean of precision and recall:

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (5.6)$$

It balances both types of classification error and is particularly useful in binary detection tasks where class imbalance is common.

Each of these metrics provides a specific perspective on model behavior, ranging from strict localization performance (IoU@0.9) to classification balance (F1 score). Together, they offer a comprehensive evaluation of the Faster R-CNN model's capacity to detect and localize cullet particles under the constraints of the proposed pipeline.

5.2 Evaluation Results for the Faster R-CNN Model

The performance of the Faster R-CNN model, configured for binary object detection (cullet vs. background), was quantitatively assessed using the evaluation metrics described previously. Table 5.2 summarizes the results computed over the entire test dataset.

Table 5.2: Faster R-CNN Performance Metrics on the Test Set

Metric	Value
Mean IoU@0.9	0.9541
Precision	0.9245
Recall	0.9228
F1 Score	0.9236
Approximate Accuracy	0.8667
False Positive Rate	0.0658
False Negative Rate	0.0727

The model achieves a mean Intersection over Union (IoU) of 0.9541 for detections with an IoU threshold of 0.9, indicating accurate spatial alignment between predicted and true bounding boxes. Precision and recall are both above 0.92, reflecting a balanced detection capability with low false detection and omission rates. The F1 score of 0.9236 further confirms this balance. An approximate accuracy of 86.67% is reported, but given that true negatives are not explicitly annotated or enumerable in object detection settings, the reported accuracy is based on partial counts and should not be interpreted as a stand-alone indicator of model effectiveness.

Table 5.3: Confusion Matrix for Test Set Predictions (Cullet vs. Background)

Ground Truth	Predicted Cullet	Predicted Background
Cullet	421 (TP)	33 (FN)
Background	32 (FP)	— (TN not tracked)

As shown in Table 5.3, the number of true positives significantly exceeds the number of false positives and false negatives. The false positive rate (6.58%) and false negative rate (7.27%) remain low, supporting the reliability of the model for automated cullet particle detection. It is important to note that true negatives (background correctly ignored) are not explicitly tracked in object detection tasks, and are therefore omitted from this matrix.

However, during automated cullet analysis at inference time, the classification model operates on image regions cropped from predicted bounding boxes produced by the detection model. As a result, detection errors directly affect the classification stage. False positives may lead to background regions or irrelevant fragments being incorrectly classified as cullet materials, while false negatives cause actual cullet particles to be excluded from classification. Although the detection pipeline applies a strict IoU threshold and includes padding to preserve contextual information, classification performance remains sensitive to the quality of the detected regions.

Comparison of Validation and Test Set Results

To assess generalization performance, the model was also evaluated on a separate validation set under the same conditions (IoU threshold of 0.9). Table 5.4 summarizes the comparative results.

Table 5.4: Comparison of Validation and Test Set Performance

Metric	Validation Set	Test Set
Mean IoU@0.9	0.9600	0.9541
Precision	0.9494	0.9245
Recall	0.9550	0.9228
F1 Score	0.9521	0.9236
Approximate Accuracy	0.9074	0.8667
True Positives (TP)	392	421
False Positives (FP)	20	32
False Negatives (FN)	20	33
False Positive Rate	0.0463	0.0658
False Negative Rate	0.0485	0.0727

The results demonstrate strong consistency across both validation and test sets. While minor performance degradation is observed on the test set—particularly in precision and recall—the overall detection capability remains stable. The F1 score decreases by only 2.85% and the mean IoU remains high in both evaluations. This indicates that the model generalizes well to unseen data without significant overfitting.

The Faster R-CNN model, based on a ResNet-50-FPN backbone and configured with a detection threshold of 0.9, exhibits high spatial accuracy and balanced detection characteristics in both validation and test scenarios. The evaluation confirms the model’s robustness for the binary classification of cullet particles and its suitability for deployment within the automated material analysis pipeline.

5.3 Evaluation Metrics for Object Classification

To evaluate the performance of the ResNet18 model used for classifying cullet particles, a set of standard classification metrics was adopted. These metrics quantify the model's ability to correctly assign material class labels to image crops obtained from the object detection step. All evaluations are performed on a per-particle basis, and results are aggregated across the dataset.

- **Accuracy:** Accuracy reflects the proportion of correctly classified instances among all predictions. This metric provides a general overview of the model's performance but can be misleading in the presence of class imbalance.

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \quad (5.7)$$

- **Precision (per class):** Precision measures the proportion of instances classified as a given class that actually belong to that class. For a class c , it is defined as:

$$\text{Precision}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FP}_c} \quad (5.8)$$

Where TP_c is the number of true positives for class c , and FP_c is the number of false positives for class c .

- **Recall (per class):** Recall measures the proportion of instances of a given class that are correctly identified:

$$\text{Recall}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FN}_c} \quad (5.9)$$

- **F1 Score (per class):** The F1 score is the harmonic mean of precision and recall, providing a single metric that balances both:

$$\text{F1}_c = 2 \cdot \frac{\text{Precision}_c \cdot \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c} \quad (5.10)$$

This metric is particularly relevant for imbalanced class distributions, as observed in some cullet datasets.

- **Confusion Matrix:** A confusion matrix summarizes the classification results by presenting the distribution of predicted versus true labels. Each row corresponds to an actual class, and each column to a predicted class. Diagonal entries represent correct classifications, while off-diagonal values indicate misclassification. The confusion matrix allows qualitative inspection of systematic errors, such as confusion between visually similar materials.

These metrics collectively provide a detailed view of the classification model's behaviour and are used to assess both general performance and class-specific behaviour in the cullet classification pipeline.

5.3.1 Classification Performance Evaluation

The performance of the ResNet-18 classification model was evaluated using standard multi-class metrics across both the training and validation datasets. These metrics include class-wise precision, recall, and F1-score, as well as overall accuracy, macro averages (unweighted across classes), and weighted averages (accounting for class frequency). Confusion matrices were also computed to inspect common misclassification patterns.

Test Results

As shown in Table 5.5, the classifier achieved an overall accuracy of 93.17% on the test dataset. High precision and recall values were obtained for the dominant classes, particularly *white glass* and *blue glass*, both achieving F1-scores above 0.98. Classes with very limited support, such as *other white glass*, *stones/ceramics/porcelain*, and *opals*, were also classified with high apparent accuracy. However, the small number of samples in these classes limits statistical reliability.

For the *green glass* and *amber glass* the model has a decay in the performance due to frequent confusion between these two classes, as shown in the Table 5.6. Also, the *organic/plastics* and *metals* categories were particularly under-represented and poorly classified.

Table 5.5: Classification Report on the Test Set

Class	Precision	Recall	F1-score	Support
White Glass	0.9783	0.9891	0.9837	274
Blue Glass	0.9773	1.0000	0.9885	86
Other White Glass	1.0000	0.8750	0.9333	8
Green Glass	0.8125	0.3939	0.5306	33
Amber Glass	0.6852	0.9024	0.7789	41
Stones/Ceramics/Porcelain	1.0000	0.8000	0.8889	5
Opals	1.0000	0.8000	0.8889	5
Plastics	0.0000	0.0000	0.0000	1
Metals	0.2500	1.0000	0.4000	1
Macro Avg	0.7448	0.7512	0.7103	–
Weighted Avg	0.9367	0.9317	0.9267	–
Accuracy	0.9317			454

Table 5.6: Confusion Matrix for ResNet-18 Classifier on Test Set

True \ Pred	White	Blue	O.W.	Green	Amber	SCP	Opals	Plast	Metal
White	271	2	0	1	0	0	0	0	0
Blue	0	86	0	0	0	0	0	0	0
O.W.	1	0	7	0	0	0	0	0	0
Green	3	0	0	13	17	0	0	0	0
Amber	0	0	0	2	37	0	0	0	2
SCP	0	0	0	0	0	4	0	0	1
Opals	1	0	0	0	0	0	4	0	0
Plast	1	0	0	0	0	0	0	0	0
Metal	0	0	0	0	0	0	0	0	1

Validation Results

On the validation set, the classifier achieved an overall accuracy of 90.29%, with a weighted F1-score of 90.42% (Table 5.7). Core glass classes such as *white glass* and *blue glass* remained robustly classified, both maintaining F1-scores above 0.94. Performance was moderately strong for *amber glass*, *green glass*, and *stones/ceramics/porcelain*, though these classes continued to show inter-class confusion, particularly between green and amber hues.

The *opals* class received no correct predictions on the validation set, and the low support for *organic/plastics* and *metals* led to unstable metrics. These outcomes underscore the challenge of generalization for under-represented classes, likely exacerbated by class imbalance and limited training samples.

Table 5.7: Classification Report on the Validation Set

Class	Precision	Recall	F1-score	Support
White Glass	0.9000	0.9863	0.9412	73
Blue Glass	0.9934	0.9868	0.9901	152
Other White Glass	1.0000	1.0000	1.0000	2
Green Glass	0.8780	0.6429	0.7423	56
Amber Glass	0.8023	0.9079	0.8519	76
Stones/Ceramics/Porcelain	0.9318	0.8367	0.8817	49
Opals	0.0000	0.0000	0.0000	0
Plastics	1.0000	0.5000	0.6667	2
Metals	0.1667	0.5000	0.2500	2
Macro Avg	0.7414	0.7067	0.7026	–
Weighted Avg	0.9146	0.9029	0.9042	–
Accuracy	0.9029			412

Table 5.8: Confusion Matrix for ResNet-18 Classifier on Validation Set

True \ Pred	White	Blue	O.W.	Green	Amber	SCP	Opals	Plast	Metal
White	72	0	0	1	0	0	0	0	0
Blue	2	150	0	0	0	0	0	0	0
O.W.	0	0	2	0	0	0	0	0	0
Green	3	0	0	36	16	1	0	0	0
Amber	0	0	0	4	69	2	0	0	1
SCP	2	1	0	0	1	41	1	0	3
Opals	0	0	0	0	0	0	0	0	0
Plast	0	0	0	0	0	0	0	1	1
Metal	1	0	0	0	0	0	0	0	1

Discussion

These results confirm the model capacity to learn robust representations for dominant material classes with adequate support. However, performance across datasets indicates a general sensitivity to class imbalance and inter-class visual similarity, particularly between *green glass* and *amber glass*. The model also demonstrates limited generalization on less represented classes, which may compromise classification reliability during inference on real-world samples.

Potential strategies to mitigate these issues include class rebalancing, targeted data augmentation, or incorporating cost-sensitive loss functions to address skewed class distributions. Despite these limitations, the model demonstrates robust performance for the primary glass categories, supporting its integration into the automated cullet analysis pipeline.

The model confusion between the classes *green glass* and *amber glass* does not have a critical downstream tasks, as these categories are merged into a unified *green/amber glass* class for mass estimation and quality assessment. In contrast, the model poor classification performance on non-glass contaminants represents a structural vulnerability in the pipeline.

5.3.2 Chan-Vese Model Performance Evaluation

The performance of the Chan-Vese segmentation algorithm is evaluated in this section based on qualitative visual analysis. Since no ground truth masks were annotated for segmentation, standard quantitative metrics could not be computed. Nevertheless, a visual inspection of the results provides meaningful insight into the model effectiveness across different cullet types.

Figure 5.1 presents the results of the Chan-Vese segmentation algorithm applied to four representative cullet particles from different material classes: white glass, amber glass, and green glass. Each example illustrates a complete segmentation pipeline, including the original image, the initial contours identified by the model, the generated binary mask, and the final filtered contour.

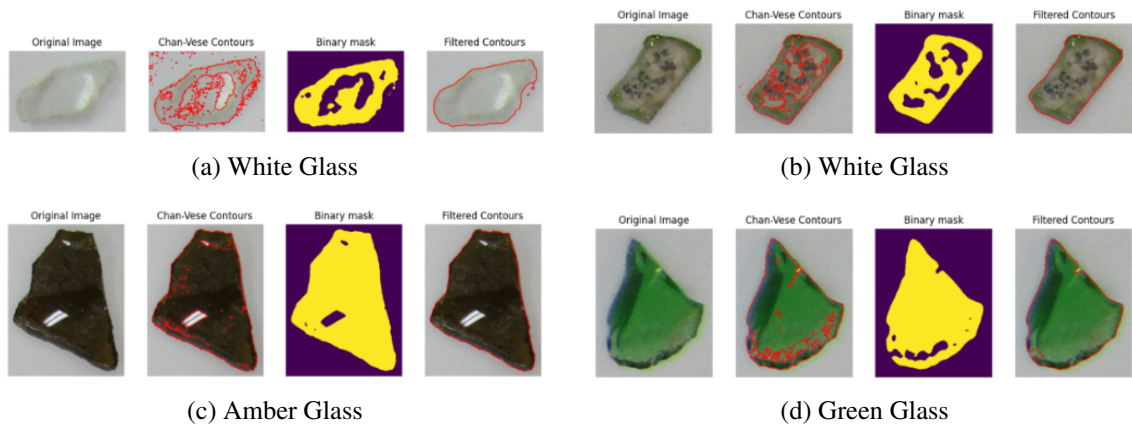


Figure 5.1: Visual examples of the Chan-Vese segmentation algorithm applied to different cullet particles. In each case, the sequence of images shows the progression from the original sample through model detected contours, followed by binary mask generation, and culminating in the final filtered segmentation result.

In the case of **white glass** (Figures 5.1a and 5.1b), the model successfully captures the primary particle boundary despite the low contrast between the object and the background. However, challenges arise when surface contamination is present. In Figure 5.1b, the cullet particle exhibits visible traces of moisture and dust, which interfere with the model ability to correctly differentiate the transparent particle from the background. Since the Chan-Vese algorithm relies on regional intensity differences to evolve contours, such localized variations in texture and colour can be incorrectly interpreted as particle boundaries.

Both the **amber glass** (Figure 5.1c) and **green glass** (Figure 5.1d) examples demonstrate the algorithm's ability to segment particles with higher visual contrast against the background. In both cases, the contour evolution proceeds in a controlled and consistent manner, with the outer boundaries of the particles accurately captured and minimal intrusion of internal false contours. However, in the amber glass sample, the algorithm tends to identify regions with strong light reflections as potential boundaries, resulting in some internal contour artifacts during early iterations. These are largely mitigated during the post-processing phase. Similarly, minor noise along the lower edge is observed in the initial contour set for the green glass particle but is also effectively corrected. The similarity in performance across these two examples suggests that the Chan-Vese algorithm is particularly well-suited to darker or more opaque cullet materials, where intensity-based separation from the background is more pronounced.

The Chan-Vese algorithm demonstrated the ability to segment cullet particles with diverse colours and surface textures. However, its performance is notably affected by noise, specular reflections, and low-contrast boundaries, conditions commonly found in transparent or highly reflective materials. In such cases, the algorithm may produce unstable or inaccurate contours. The post-processing stage played a crucial role in mitigating these limitations, enabling more reliable boundary extraction and overall acceptable segmentation accuracy.

Chapter 6

Conclusions

This thesis set out to address the need for automation in the quality assessment of recycled glass (cullet) by pursuing two primary objectives. The first was to develop a deep learning architecture capable of detecting and classifying individual particles within cullet sample images, distinguishing between recyclable glass and contaminants across nine predefined classes. The second objective was to automate the assessment of sample purity by analysing the estimated class-wise weight distribution.

The proposed system successfully fulfilled its intended goals through the integration of three core components. A Faster R-CNN model, fine-tuned for binary detection, was used to accurately identify cullet particles in realistic industrial conditions, achieving a mean IoU of 0.9541 and an F1 score of 0.9236 on the test set. The ResNet-18 classifier reliably categorized the detected particles into nine material classes, attaining a weighted F1 score of 0.93 on the test set. Finally, the Chan–Vese segmentation algorithm was employed to estimate the area of each individual particle, which was then converted into a corresponding weight using class-specific density values obtained from the training set. This enabled an image-based estimation of the class-wise weight distribution, supporting automated quality analysis. Collectively, these components constitute the pipeline for cullet quality analysis based solely on visual data.

Several technical and operational challenges emerged during the development of the system. A primary issue was the imbalance in the dataset, with classes such as metals and plastics significantly under-represented. This limited the model ability to generalize and consistently recognize rare but industrially critical contaminants. Early experiments with multi-class object detection further revealed degraded localization performance, prompting a shift to a modular architecture: a binary detector for particle identification followed by a separate classifier for material categorization. While this design improved robustness, it introduced inter-stage dependencies and additional computational overhead. The system also faced persistent difficulties in distinguishing visually similar materials, particularly between green and amber glass, despite their being aggregated during final mass estimation. Overall, limited classification performance on low-frequency and visually ambiguous classes remains a key constraint of the current solution.

6.1 Future Work

The limitations observed throughout this project suggest several promising directions for future research and system enhancement. First, expanding and rebalancing the dataset, particularly by increasing the representation of under-represented contaminant classes, would significantly improve the system robustness. This could be achieved through additional data collection, synthetic data generation, or the application of semi-supervised learning techniques to reduce annotation overhead.

With a more comprehensive dataset in place, it would be worthwhile to revisit multi-class object detection. This could involve designing a specialized architecture tailored to the cullet classification task, potentially integrating an enhanced feature extraction pipeline that leverages sensor data to improve inter-class discrimination.

Ensuring generalization beyond controlled laboratory conditions is also essential. Domain adaptation and transfer learning strategies should be explored to facilitate deployment in diverse industrial environments with varying imaging conditions.

The segmentation stage could benefit from replacing the Chan–Vese model with deep learning-based techniques such as Mask R-CNN or U-Net, which offer improved accuracy and efficiency, particularly in scenarios involving complex or overlapping particle configurations. However, implementing these architectures would require the creation of new annotations specifically tailored for the segmentation task.

For practical deployment, real-time performance must be addressed. This includes integrating the system with industrial hardware, optimizing inference speed, and handling challenges such as motion blur and lighting variation on conveyor systems.

6.2 Final Remarks

In summary, this work demonstrated a novel AI-driven approach for the real-time evaluation of recycled glass quality, aligning with sustainability goals and advancing industrial automation. The system lays a strong foundation for further research, with future work focused on improving robustness, scalability, and deployment readiness.

References

- [1] F. Testa, F. Iraldo, A. Vaccari, and E. Ferrari, “Why eco-labels can be effective marketing tools: Evidence from a study on italian consumers: Why eco-labels can be effective marketing tools,” *Business Strategy and the Environment*, vol. 24, Nov. 2013. DOI: [10.1002/bse.1821](https://doi.org/10.1002/bse.1821).
- [2] BA Glass Group, *Ba glass annual report 2023: Sustainability and innovation*, 2023.
- [3] J. Zarzycki, *Glasses and the vitreous state*. Cambridge university press, 1991.
- [4] J. E. Shelby, *Introduction to Glass Science and Technology*. Royal Society of Chemistry, 2020.
- [5] J. C. Mauro, Y. Yue, A. J. Ellison, P. K. Gupta, and D. C. Allan, “Viscosity of glass-forming liquids,” *Proceedings of the National Academy of Sciences*, vol. 106, no. 47, pp. 19 780–19 784, 2009.
- [6] R. Zallen, *The physics of amorphous solids*. John Wiley & Sons, 2008.
- [7] A. Paul, *Chemistry of glasses*. Springer Science & Business Media, 1989.
- [8] S. C. Rasmussen, *How glass changed the world: The history and chemistry of glass from antiquity to the 13th century*. Springer Science & Business Media, 2012.
- [9] F. T. Wallenberger and J. D. MacKenzie, *Modern Glass Science and Technology*. Springer, 2009. DOI: [10.1007/978-1-4419-0587-0](https://doi.org/10.1007/978-1-4419-0587-0).
- [10] S. R. Scholes, “Modern glass practice,” (*No Title*), 1975.
- [11] Bucher Emhart Glass, *Container defect causes and remedies*, Accessed July 12, 2025, Bucher Emhart Glass, 2025. [Online]. Available: <https://www.emhartglass.com/sites/default/files/publications/2020-09/BR0060%20-%20Container%20defect%20causes%20and%20remedies.pdf>.
- [12] Bucher Emhart Glass. “Glass container production process.” Accessed: 2025-07-12. (2025), [Online]. Available: <https://www.emhartglass.com/node/26#glass-containers-10>.
- [13] P. P. Mohit Sewak Md. Rezaul Karim, *Deep Learning for Vision Systems*. Manning Publications, 2020.
- [14] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, “Object detection in 20 years: A survey,” *Proceedings of the IEEE*, vol. 111, no. 3, pp. 257–276, 2023.
- [15] P. Viola and M. J. Jones, “Robust real-time face detection,” *International journal of computer vision*, vol. 57, no. 2, pp. 137–154, 2004.
- [16] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR’05)*, Ieee, vol. 1, 2005, pp. 886–893.

- [17] P. Modi, *Convolutional neural networks for dummies: A step-by-step cnn tutorial*, 2020. [Online]. Available: <https://medium.com/@prathammodi001/convolutional-neural-networks-for-dummies-a-step-by-step-cnn-tutorial-e68f464d608f>.
- [18] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2014, pp. 580–587.
- [19] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.
- [20] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.
- [21] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.
- [22] W. Liu, D. Anguelov, D. Erhan, *et al.*, "Ssd: Single shot multibox detector," in *European conference on computer vision*, Springer, 2016, pp. 21–37.
- [23] D. Todorovic, "Gestalt principles," *Scholarpedia*, vol. 3, no. 12, p. 5345, 2008.
- [24] J. MacQueen, "Multivariate observations," in *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, 1967, pp. 281–297.
- [25] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 5, pp. 603–619, 2002.
- [26] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International journal of computer vision*, vol. 59, no. 2, pp. 167–181, 2004.
- [27] S. Beucher, *The watershed transformation*, Accessed: 2025-07-15, 2021. [Online]. Available: <https://people.cmm.minesparis.psl.eu/users/beucher/wtshed.html>.
- [28] L. Vincent and P. Soille, "Watersheds in digital spaces: An efficient algorithm based on immersion simulations," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 13, no. 06, pp. 583–598, 1991.
- [29] OpenCV Contributors, *Image segmentation with watershed algorithm*, Accessed: 2025-07-15. [Online]. Available: https://docs.opencv.org/4.x/d3/db4/tutorial_py_watershed.html.
- [30] T. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models-their training and application," *Comput. Vis. Image Underst.*, vol. 61, pp. 38–59, 1995.
- [31] T. Cootes, G. Edwards, and C. Taylor, "Active appearance models," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, pp. 681–685, Jul. 2001. DOI: 10.1109/34.927467.
- [32] D. Cremers, T. Kohlberger, and C. Schnörr, "Nonlinear shape statistics in mumford-shah based segmentation," in *Computer Vision – ECCV 2002*, A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, Eds., ser. Incs, Springer Verlag, vol. 2351, Springer Verlag, 2002, 93–108.

- [33] A. Tsai, A. Yezzi, W. Wells, *et al.*, “A shape-based approach to the segmentation of medical imagery using level sets,” *IEEE transactions on medical imaging*, vol. 22, no. 2, pp. 137–154, 2003.
- [34] H. Kaiming, G. Georgia, D. Piotr, and G.-s. Ross, “Mask r-cnn,” in *Proceedings of the IEEE international conference on computer vision*, vol. 2017, 2017, pp. 2961–2969.
- [35] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical image computing and computer-assisted intervention*, Springer, 2015, pp. 234–241.
- [36] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 11, pp. 1330–1334, 2002.
- [37] G. Bradski, “The opencv library,” *Dr. Dobbs’s Journal: Software Tools for the Professional Programmer*, vol. 25, no. 11, pp. 120–123, 2000.
- [38] T.-Y. Lin, M. Maire, S. Belongie, *et al.*, “Microsoft coco: Common objects in context,” in *European conference on computer vision*, Springer, 2014, pp. 740–755.
- [39] K. J. Zuiderveld, “Contrast limited adaptive histogram equalization,” in *Graphics Gems*, P. S. Heckbert, Ed., Elsevier, 1994, pp. 474–485, ISBN: 9780123361561. DOI: [10.1016/b978-0-12-336156-1.50061-6](https://doi.org/10.1016/b978-0-12-336156-1.50061-6). [Online]. Available: <https://doi.org/10.1016/b978-0-12-336156-1.50061-6>.
- [40] T.-Y. Lin, M. Maire, S. Belongie, *et al.*, “Microsoft coco: Common objects in context,” in *European conference on computer vision*, Springer, 2014, pp. 740–755.
- [41] D. Masters and C. Luschi, “Revisiting small batch training for deep neural networks,” *arXiv preprint arXiv:1804.07612*, 2018.
- [42] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, “On large-batch training for deep learning: Generalization gap and sharp minima,” *arXiv preprint arXiv:1609.04836*, 2016.
- [43] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.
- [44] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.
- [45] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [46] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE conference on computer vision and pattern recognition*, Ieee, 2009, pp. 248–255.
- [47] I. Goodfellow, *Deep learning*, 2016.
- [48] L. N. Smith, “A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay,” *arXiv preprint arXiv:1803.09820*, 2018.
- [49] T. F. Chan and L. A. Vese, “Active contours without edges,” *IEEE Transactions on image processing*, vol. 10, no. 2, pp. 266–277, 2001.

- [50] C. Li, C. Xu, C. Gui, and M. D. Fox, “Distance regularized level set evolution and its application to image segmentation,” *IEEE transactions on image processing*, vol. 19, no. 12, pp. 3243–3254, 2010.

Appendix A

Cullet Treatment Process Diagram

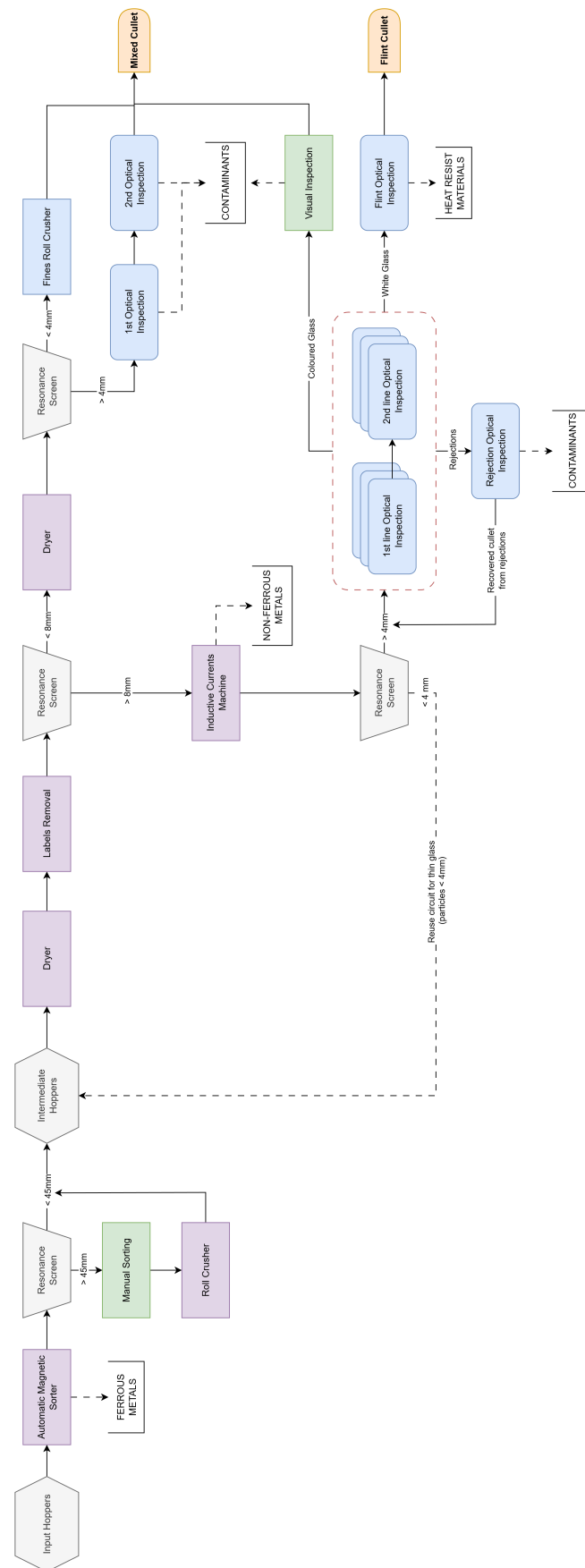


Figure A.1: Detailed flowchart of the cullet treatment process. The vertical layout illustrates all unit operations and material pathways used in transforming post-consumer recycled glass into furnace-ready cullet.