NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTMENT OF MECHANICAL AND INDUSTRIAL ENGINEERING

CLÁUDIA OLIVEIRA CORREIA

BSc in Mechanical Engineering

DETECTION OF WELD SURFACE DEFECTS WITH COMPUTER VISION TECHNIQUES

MASTER IN MECHANICAL ENGINEERING

NOVA University Lisbon

November, 2025

NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTMENT OF MECHANICAL AND INDUSTRIAL ENGINEERING

DETECTION OF WELD SURFACE DEFECTS WITH COMPUTER VISION TECHNIQUES

CLÁUDIA OLIVEIRA CORREIA

BSc in Mechanical Engineering

Adviser: Nuno Alberto Marques Mendes
Assistant Professor, NOVA School of Science and Technology

Examination Committee

Chair: Telmo Jorge Gomes dos Santos
Full Professor, NOVA School of Science and Technology

Rapporteur: Pedro Manuel Cardoso Vieira
Associate Professor, NOVA School of Science and Technology

Adviser: Nuno Alberto Marques Mendes
Assistant Professor, NOVA School of Science and Technology

MASTER IN MECHANICAL ENGINEERING

NOVA University Lisbon
November, 2025

DETECTION OF WELD SURFACE DEFECTS WITH COMPUTER VISION TECHNIQUES

This document was created with the (pdf/Xe/Lua)LaTeX processor and the NOVAthesis template (v7.2.1) [1].

Acknowledgements

I would like to begin by expressing my sincere gratitude to Professor Nuno Mendes and Professor Pedro Vieira for their invaluable guidance, support, and availability throughout this dissertation. Their mentorship helped me discover an area I now find both fascinating and highly relevant. I am also grateful to Mr. António Campos and Mr. Paulo Magalhães for their patience, support, and assistance during this study.

I am deeply grateful to the NOVA School of Science and Technology, in particular the Department of Mechanical and Industrial Engineering, for providing me with the knowledge and skills that have shaped my academic and professional journey. I owe special appreciation to the professors of the department, whose teaching and inspiration showed me how engaging and rewarding engineering can be.

To my colleagues over these past five years, I am grateful for the friendships and shared experiences that enriched my time at university. In particular, I would like to acknowledge my closest group of friends—you know who you are—whose presence has marked my life in ways I cannot fully express, and with whom I share unforgettable memories. I am also grateful to my friends beyond university, especially my two closest friends, who I can always rely on regardless of time or circumstance. Your advice, emotional support, and patience in listening to me have been invaluable.

I would also like to recognize my two school teachers who inspired me to pursue this career and contributed greatly to the person I am today.

Lastly, I wish to express my profound gratitude to my family. To my parents, for their unconditional love and support in all my decisions, even when skeptical at first, and for enabling me to follow the academic path I desired. To my brother, for always being there for me. To my cousin Marta, who has been like an older sister, for her continuous support and care. To my godfather, for motivating me to be better and to think outside the box—words that will stay with me throughout life. To my grandmothers, for their unconditional love and support in every way. And to my uncle and aunts, for always standing by me.

This work was developed under the project "Agenda Drivolution - Transição para a fábrica do futuro" (C644913740-00000022) - Aviso 2021-C05i01-01 PRR, funded by IAPMEI

- Agência para a Competitividade e Inovação, I. P., within the scope of the Portuguese "Plano de Recuperação e Resiliência (PRR)" / União Europeia - NextGenerationEU.
https://www.fct.unl.pt/investigacao/projetos-financiados/drivolution-transicao-para-fabrica-do-futuro

Abstract

Inspection of welding seams is a critical task in industrial production, as undetected defects can compromise both safety and product quality. This thesis investigates computer vision techniques for the automatic detection and classification of welding surface defects, with the broader objective of identifying the most suitable combination of hardware and methodology for developing an automated visual inspection system.

To support this work, a dedicated dataset of weld images was acquired using a custom image acquisition system and preprocessed with specialized techniques. Multiple deep learning models were evaluated for both detection and classification tasks, with particular attention to the effects of data augmentation, class imbalance, and data leakage. Three inspection pipelines were designed and compared: a detection–classification pipeline, a classification-only pipeline, and a hybrid pipeline combining both strategies.

The results demonstrated that defect detection based on models such as YOLOv8n, combined with convolutional architectures such as ConvNeXt, achieved consistent performance. Hybrid pipelines improved detection for minority defect classes, although limitations persisted for extremely rare categories.

The study demonstrates that careful dataset design and augmentation strategies are essential for achieving reliable performance, and that detection-based approaches provide a stronger foundation for automated weld inspection. These findings contribute to the development of a robust computer vision system capable of reducing manual inspection effort and improving quality assurance in industrial welding processes.

Keywords: Welding surface defects, Computer vision, Visual inspection, Detection, Classification, Deep learning

Resumo

A inspeção de soldaduras constitui uma etapa fundamental na produção industrial, dado que defeitos não detetados podem comprometer tanto a segurança como a qualidade dos produtos. Neste contexto, a presente dissertação concentra-se no desenvolvimento de um sistema de visão capaz de detetar e classificar automaticamente defeitos superficiais em soldaduras, recorrendo a técnicas de visão computacional.

Para tal, foi concebido um sistema de aquisição de imagens específico para este tipo de inspeção. As imagens recolhidas foram posteriormente submetidas a um pré-processamento adequado, com o objetivo de melhorar a sua qualidade e uniformidade. Além disso, foram avaliados diversos modelos de aprendizagem profunda, sendo considerado também o impacto de estratégias como aumento de dados, balanceamento de classes e prevenção de fugas de dados nos processos de deteção e classificação.

Foram exploradas três abordagens de inspeção: uma pipeline de deteção e classificação, uma dedicada exclusivamente à classificação e uma solução híbrida que integra ambas as estratégias.

Os resultados demonstraram que as pipeline de deteção e classificação de defeitos, utilizando modelos como o YOLOv8n em combinação com arquiteturas convolucionais como o ConvNeXt, apresentam os melhores desempenhos. Por outro lado, as soluções híbridas podem melhorar a deteção de classes minoritárias, apesar de ainda apresentarem algumas limitações.

Este estudo evidencia a importância de um sistema de aquisição de imagens de elevada qualidade, de um conjunto de dados cuidadosamente estruturado, de técnicas de pré-processamento apropriadas e de uma seleção criteriosa de modelos como fatores essenciais para o sucesso de sistemas automáticos de inspeção visual.

Palavras-chave: Defeitos superficiais em soldaduras, Visão computacional, Inspeção visual, Deteção, Classificação, Aprendizagem profunda

Contents

List of Figures

List of Tables

1

Introduction

1.1 Motivation

Welded structures play a critical role in many sectors, including construction, manufacturing, and transportation. Welds are essential for the structural integrity of machines, buildings, and vehicles. Defects in welds can compromise mechanical performance, shorten component lifespan, and, in severe cases, pose significant safety risks. Consequently, thorough inspection of welds is vital to ensure quality and reliability.

Non-Destructive Testing (NDT) methods offer an effective means of evaluating weld quality, enabling the detection and characterization of defects without damaging the components. NDT includes a variety of techniques such as thermal, acoustic, optical, electrical, and magnetic methods. Among these, visual inspection is particularly common due to its simplicity, low cost, and practical applicability in industrial environments.

Despite its widespread use, traditional visual inspection has notable limitations. It is time-consuming, heavily reliant on the skill and experience of the inspector, and susceptible to human error. Maintaining consistent inspection quality across large-scale production, where numerous welds must be evaluated rapidly and accurately, is particularly challenging. These limitations highlight the need for reliable, automated inspection methods capable of maintaining high-quality standards efficiently and consistently.

Recent advances in computer vision and artificial intelligence (AI) have facilitated the development of automated visual inspection systems. These systems utilize high-resolution imaging, image processing, and machine learning and deep learning algorithms to detect, localize, and classify weld defects with high accuracy.

Automated visual inspection has become an essential component of modern manufacturing, offering significant advantages such as increased inspection speed, improved consistency, reduced waste, and enhanced overall product reliability. Furthermore, these systems have broad applicability, from small-scale production to large industrial operations, making them versatile tools for quality assurance.

Nevertheless, challenges remain in implementing automated weld inspection systems.

Welds can vary in appearance due to differences in materials, welding techniques, environmental conditions, and production settings. Consequently, developing robust algorithms capable of handling diverse defect types and real-world variations remains a key research focus.

In summary, ensuring the quality of welded structures is critical for safety, performance, and reliability. While traditional NDT methods remain valuable, automated visual inspection systems powered by computer vision and AI present a promising solution for overcoming the limitations of manual inspection. This dissertation aims to contribute to this area by developing and evaluating a comprehensive system for the automatic detection and classification of surface defects in welds.

1.2 Objectives

This dissertation aims to identify high performance hardware and methodology for developing a vision system capable of automatically detecting surface defects in welds. By applying advanced image acquisition methods, pre-processing techniques, and state-of-the-art algorithms, it seeks to determine the most effective combination.

To achieve this, a case study is conducted, focusing on the inspection of a welded metal component.

1.3 Structure of the dissertation

This dissertation is structured into seven chapters, each containing relevant subchapters.

Chapter 1 introduces the motivation, objectives, and structure of the proposed work.

Chapter 2 presents a literature review of essential concepts for this work, including surface defects in welds, non-destructive techniques, automated visual inspection systems, computer vision techniques, and model architectures.

Chapter 3 describes the equipment and procedures used to acquire images of the welded component for the development of the dataset.

Chapter 4 details the development of the dataset required for the defect recognition system, including image pre-processing and exploratory data analysis, as well as dataset preparation steps.

Chapter 5 details the models and metrics employed in the pipeline's development, from model testing and ensemble strategies to detection and classification approaches.

Chapter 6 analyzes the results obtained from the developed pipelines, assessing the performance and validity of the implemented methods.

Finally, Chapter 7 summarizes the conclusions of this work, emphasizing the achieved objectives and suggesting directions for future improvements.

2

Literature Review

Welding is a critical process in many industries, including shipbuilding, automotive manufacturing, aerospace, and energy production. The integrity of welds directly affects structural performance, safety, and the reliability of the final product. However, welding is prone to defects such as cracks, porosity, undercuts, and lack of fusion, which can compromise component performance. Traditional defect detection methods, including visual inspection, ultrasonic testing, and radiography, have been widely used for decades but are limited by the inspector's skill and difficulty detecting subsurface defects, particularly in complex geometries or hard-to-reach areas. Furthermore, the rise of Industry 4.0, characterized by cyber-physical systems, IoT, automation, and data analytics, has introduced new technologies capable of revolutionizing welding inspection and quality control [2].

Computer vision is a rapidly evolving field that enables machines to interpret and analyze visual data. In industrial contexts, it plays a central role in automation and quality control. A particularly important application is Automated Visual Inspection (AVI), which enhances manufacturing efficiency by replacing manual inspection with automated defect detection and quality assessment.

Machine vision systems, as a cornerstone of modern manufacturing, combine high precision, speed, and efficiency with the advantage of non-contact measurements, making them ideally suited for integration into production lines [3]. Among their many applications—such as process control, part identification, and robotic guidance—AVI remains the most significant and widely adopted [4].

Traditional AVI algorithms typically involve two stages: feature extraction and defect identification. They rely heavily on human-designed features and are sensitive to variations in operating conditions [5].

The emergence of deep learning algorithms has significantly improved AVI by increasing defect classification accuracy and reducing reliance on handcrafted features. However, deep learning has relatively few applications in AVI to date, likely due to its dependence on large training datasets, whereas surface defect datasets are generally small and difficult to collect or label. Despite this, deep learning-based methods can automatically learn

high-level features from training data, making them more versatile for detecting diverse defect types and less sensitive to variations in conditions [6].

This chapter reviews the fundamental principles of computer vision and AVI, their significance in industrial applications, and also the challenges faced in implementing these systems.

2.1 Surface Defects in Welds

Welded joints are susceptible to a range of imperfections that may compromise structural integrity of a component. These flaws, depending on their type, size, and location, are classified as defects if they significantly impair the safety or performance of the welded structure [7, 8]. Monitoring weld integrity, therefore, requires careful examination of the fusion zone and heat-affected zone (HAZ), as defects in these areas strongly influence the joint's reliability and determine the effectiveness of detection methods [8].

Common welding defects include:**lack of penetration**, where the weld metal does not fully fill the joint; **lack of fusion**, caused by incomplete bonding between weld passes or with the base metal; **undercutting**, which forms grooves along bead edges that act as stress concentrators; **porosity**, resulting from trapped gas that reduces weld density; **cracks**, which may propagate under stress; **slag inclusions**, consisting of non-metallic trapped particles; and **overlap**, where molten metal flows over the base material without proper fusion. In contrast, ideal welds exhibit uniform penetration and complete fusion, serving as benchmarks for defect-free quality. [7–9]. Figure 2.1 illustrates typical weld defects.

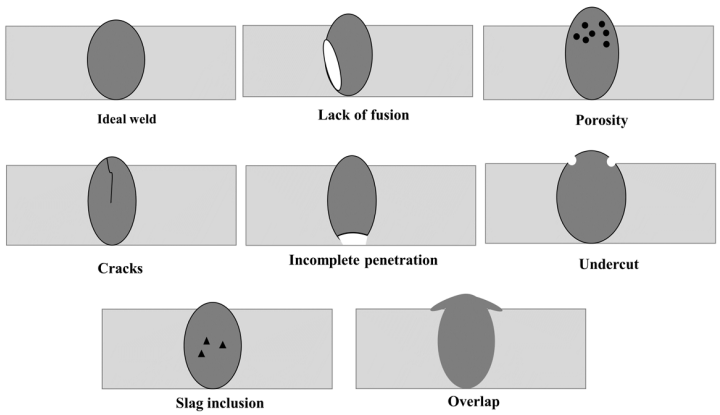


Figure 2.1: Schematic representation of various welding defects (adapted from [8]).

Understanding defect occurrence, causes, and preventive measures is essential for both manual and automated inspection, as it informs the appropriate choice of non-destructive testing (NDT) methods and supports the development of robust defect detection systems.

2.2 Non-Destructive Testing Techniques

Non-Destructive Testing (NDT) encompasses a set of methods used to evaluate the integrity of materials and detect surface or internal defects without altering the serviceability of the tested component [10]. Unlike destructive testing, NDT preserves structural integrity, making it indispensable in safety-critical industries such as aerospace, nuclear power, defense, and oil and gas pipelines.

In welding, NDT methods detect imperfections ensuring defects are identified before components enter service or during service [11]. These techniques can be categorized based on the type of defects they target and the principles they rely on, each with specific capabilities and limitations.

Visual Inspection (VI)

Visual inspection is the most straightforward and cost-effective method for weld quality assessment. It involves a visual survey of the weld bead to identify macroscopic imperfections such as undercuts, surface cracks, slag inclusions, or porosity [10, 11]. Visual inspection relies on human perception or, in automated systems, image sensors and processors. Automated systems can save time and improve detection accuracy compared to manual inspection.

Although widely used, visual inspection is limited to surface-visible defects.

Dye Penetrant Testing (DPT)

To enhance surface defect detection, visual inspection is often complemented with dye penetrant testing. DPT employs capillary action to draw fluorescent or visible dyes into surface-breaking cracks as small as 0.1 μm [11]. The procedure typically involves three steps:

1. **Penetrant application:** Ensuring full coverage of the surface.
2. **Cleaning:** Removing excess penetrant to avoid false indications.
3. **Developer application:** Drawing out penetrant trapped in flaws to make defects visible for assessment [8].

Figure 2.2 shows a typical DPT setup.

Magnetic Particle Inspection (MPI)

MPI is used on ferromagnetic materials to detect surface and near-surface defects. Ferromagnetic particles are applied over a magnetized weld surface; discontinuities disrupt magnetic flux lines, causing particle accumulation at defect sites and revealing flaws [11]. MPI, like DPT, is restricted to surface and near-surface defects.

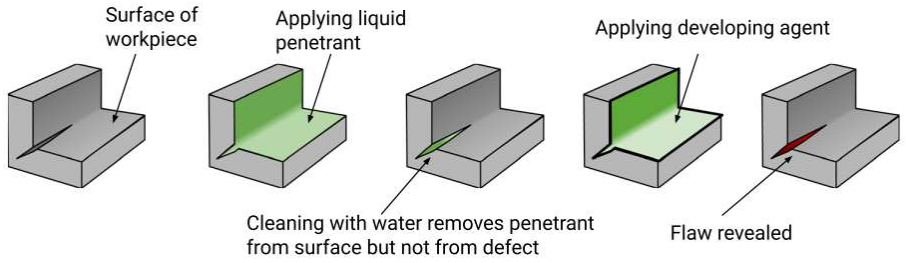


Figure 2.2: Liquid penetrant testing of manufactured components for identification of surface defects [11].

Ultrasonic testing (UT)

For subsurface flaw detection, ultrasonic testing (UT) is commonly employed. In this method, ultrasonic waves are sent into the weld. Discontinuities can reflect or scatter part of the energy, and the returned signals are analyzed to determine the presence and location of defects. UT is a versatile technique, though its effectiveness can be influenced by the material's microstructure [11].

Infrared thermography (IRT)

IRT is a non-contact technique based on heat transfer properties. Defective regions exhibit different thermal absorption and dissipation compared to sound areas, enabling detection of surface and near-surface flaws via thermal imaging (Figure 2.3) [11].

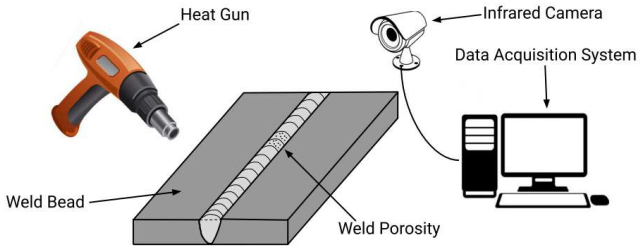


Figure 2.3: Experimental setup for weld defect detection using IR thermography [11].

Eddy current testing (ECT)

ECT uses electromagnetic induction to detect flaws in conductive materials. Cracks or porosity alter eddy current flow, which is measured by a sensing coil to identify defects [11].

Radiography

Radiography remains one of the most reliable methods for internal defect detection, particularly for porosity and cracks. X-rays or gamma rays pass through the weld, with defects appearing as darker regions on radiographic films or digital detectors. Although

it provides permanent quality records, radiography poses safety hazards and requires careful alignment for thin crack detection (Figure 2.4) [10, 11].

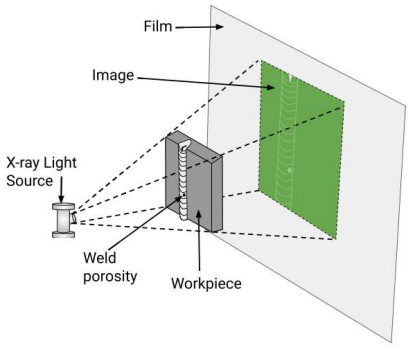


Figure 2.4: Schematic representation of experimental setup for recording radiographic images of weld bead [11].

Acoustic Emission (AE)

Acoustic emission testing monitors stress waves emitted by crack propagation or other defect activity. Unlike most NDT methods, AE enables continuous, real-time monitoring during manufacturing and service, making it effective for early defect detection and structural health monitoring.

While all methods are valuable, they are usually applied post-welding, meaning defective parts can only be reworked or discarded, leading to material and energy losses [11]. This limitation motivates the development of in-situ and automated defect detection systems, discussed in the following sections.

2.3 Automated Visual Inspection Systems

Historically, visual inspection was carried out manually by trained personnel, but this approach is monotonous, prone to human error, and lacks reproducibility [3]. Manual inspection also suffers from inefficiency, particularly for fast-moving production lines or large surface areas [12]. Studies as early as the 1970s reported human error rates in manual inspection tasks reaching 20–30% due to fatigue and attention deficits [13]. These limitations motivated the exploration of automated solutions, initially as decision-support tools and later as fully autonomous inspection systems [12].

Since the 1970s, Automated Visual Inspection (AVI) systems have gained traction due to advancements in computing power and camera technology. They are now widely employed in industries such as electronics, automotive, aerospace, steel, semiconductor, and textiles, ensuring product reliability and safety [14]. Adoption varies by sector, with steel and semiconductor manufacturing relying heavily on AVI due to stringent quality

control requirements [6]. In steel production—critical for automotive, shipbuilding, and aerospace sectors—vision-based surface defect inspection systems are essential [15].

Despite these advantages, AVI systems face challenges including high implementation costs, the need for extensive training datasets, and variations in defect appearances. Traditional rule-based methods often struggle with complex or variable defects, motivating the adoption of deep learning approaches for improved adaptability and robustness [16]. Vision transformer models have further advanced performance, though they require significant computational resources. Notably, deep-learning breakthroughs typically see a lag of about three years before adoption in industrial AVI applications [12].

Since the late 20th century, continuous innovations in AVI have improved defect detection accuracy. For example, in 1995, POSCO and Parsytec developed an AVI system for rolling steel strip production. Subsequent advances enhanced high-speed image processing, data transmission, and illumination techniques.

AVI is employed across industries for a wide range of applications [12]:

- **Quality Inspection**: Assessing product conditions, including fabric or leather quality;
- **Damage Detection**: Identifying defects such as cracks, scratches, and dents on manufactured parts;
- **Crack Detection**: Specializing in materials like concrete or pavement;
- **Completeness Checks**: Ensuring all necessary components are present, such as rail fasteners;
- **Other Applications**: Including plant disease detection and object classification.

While AVI spans diverse applications, its role in welding is particularly crucial given the safety-critical nature of welded structures, where reliable defect detection is indispensable. Figure 2.5 illustrates a typical visual surface detection system, comprising an image acquisition unit, image processing unit, data management, and human–computer interface.

An AVI system typically consists of three main stages [3, 15, 17–19]:

1. **Image Acquisition** – Optical systems, including cameras with CCD or CMOS sensors and specialized lighting, to capture detailed images of the target surface.
2. **Image Pre-processing and Feature Extraction** –Techniques to improve quality, remove noise, and extract relevant features for analysis.
3. **Defect Detection and Classification** – Machine learning and deep learning models that analyze extracted features to detect and classify defects, supporting real-time quality control.

The following subsections discuss each of these stages in detail.

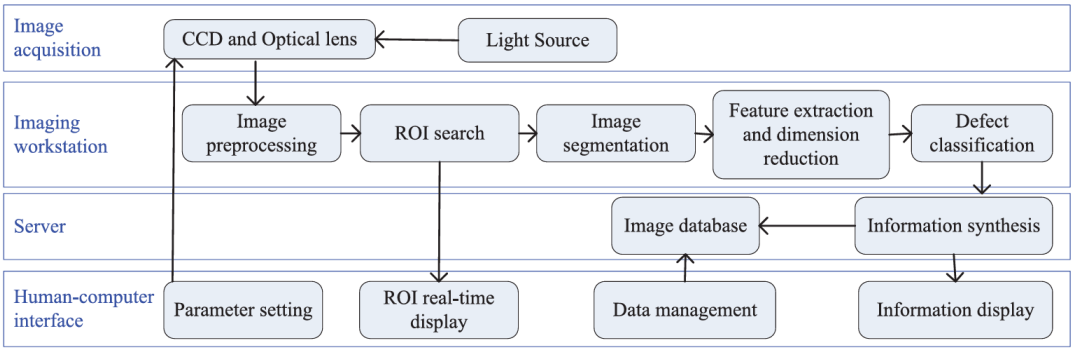


Figure 2.5: Composition of Visual Inspection System [15].

2.3.1 Image Acquisition

Typical machine vision systems consist of the structure shown in Figure 2.6, which is to be read bottom-up. This figure is deliberately shaped like a pyramid based on the process of image acquisition. The quality of image acquisition determines the reliability of the entire inspection process, as information not captured at this stage is difficult or impossible to recover in later processing [3].

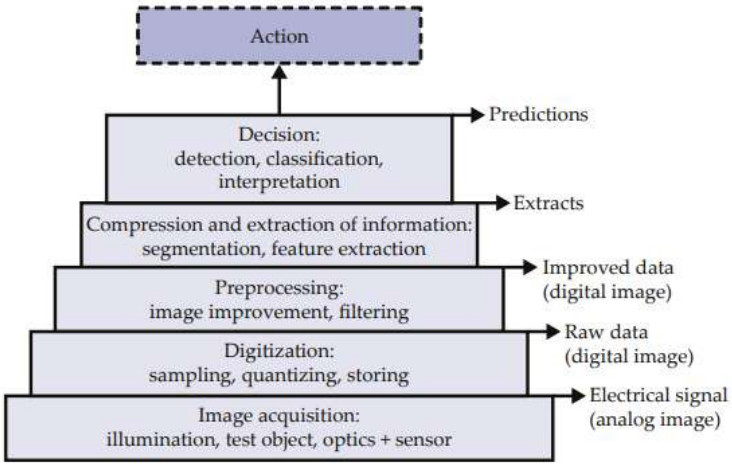


Figure 2.6: The processing chain of machine vision systems [3].

The development of image acquisition technologies has played a crucial role in the evolution of AVI systems. The development of Charge-Coupled Device (CCD) and Complementary Metal-Oxide Semiconductor (CMOS) sensors in the early 1970s enabled digital image capture for computer-based analysis [14]. CCD sensors offer high sensitivity, resolution, and low distortion, whereas CMOS sensors provide cost-effectiveness, power efficiency, and integrated preprocessing functionalities [15]. The choice of sensor is crucial for inspection performance.

Lighting also plays a key role. Fluorescent, halogen, xenon lamps, and LEDs are common sources, with LEDs now preferred due to long lifespan, low power consumption,

and uniform illumination [15].

AVI systems employ different imaging techniques, including line-scan and area-scan imaging. Line-scan cameras provide high-frequency scanning for continuous defect detection but require uniform illumination and controlled movement. Area-scan cameras capture two-dimensional images with higher resolution but are limited in large fields of view [15]. Multi-camera setups often overcome these limitations.

Advanced imaging techniques, including structured light, active triangulation, photometric stereo, 3D scanning, laser vision, and AI-driven cameras, enhance defect capture and adaptability [15].

In welding inspection, common sensors include CCD/CMOS cameras, radiography, infrared thermography, and 3D scanning. CMOS sensors dominate due to high resolution, dynamic range, and adaptability, while radiography remains essential for internal defects but requires radiation safety precautions [6, 17].

High-quality image acquisition is foundational to AVI systems, directly influencing defect detection accuracy and overall performance.

2.3.2 Image Pre-processing and Feature Extraction

After acquisition, images undergo a series of pre-processing steps to enhance quality and make defect regions more distinguishable. Pre-processing reduces noise, corrects illumination, and highlights relevant patterns, facilitating reliable feature extraction [17].

Image Pre-processing

Common pre-processing operations include image masking, spatial filtering, geometric transformations, and denoising [17]. Masking defines regions of interest (ROI), focusing computational resources and reducing inspection time. Geometric transformations (scaling, rotation, translation) correct distortions.

Noise reduction is performed via spatial filters (median, Gaussian) or transform-domain techniques (Fourier, wavelet). Advanced approaches include low-rank approximations, nuclear norm minimization, and deep learning-based denoising [17].

Illumination correction methods, such as histogram equalization, Retinex, and homomorphic filtering, normalize lighting and improve contrast. Segmentation techniques—thresholding, edge detection (Sobel, Canny, LoG), region growing, and clustering (K-means, fuzzy C-means)—separate defect regions from the background [15].

Feature Extraction and Selection

Feature extraction converts processed images into numerical descriptors that characterize defects or regions of interest [18]. Feature categories include:

- **Texture Features**: Methods such as Grey-Level Co-occurrence Matrix (GLCM), Local Binary Patterns (LBP), and Histograms of Oriented Gradients (HOG) provide

statistical descriptions of surface patterns. They are simple, fast, and generally invariant to rotation or illumination, but they are limited to low-complexity patterns and can be sensitive to noise.

- **Transform-Domain Features**: Techniques like Fourier, Gabor, and Wavelet transforms capture frequency and multi-resolution information, making them effective for periodic or multi-scale defect patterns. However, Fourier analysis lacks spatial localization, while Gabor and Wavelet transforms require significant computational resources.

- **Shape Features**: Both region-based descriptors (e.g., area, compactness) and contour-based measures (e.g., curvature, moments) are used to characterize defect geometry. Their accuracy, however, depends strongly on segmentation quality.

- **Grey-Level Features**: Intensity-based measures such as mean, variance, entropy, or histograms provide simple statistical insight into pixel distributions, though they may struggle with complex textures.

- **Model-Based Features**: Approaches such as Markov Random Fields (MRF) and autoregressive models capture local structural dependencies with relatively low computational cost. Their effectiveness, however, depends on accurate parameterization and model fitting.

- **Learning-Based Approaches**: Traditional classifiers such as Support Vector Machines (SVMs), Artificial Neural Networks (ANNs), Random Forests, or clustering algorithms like K-means are widely used. They are effective when supplied with discriminative features, but performance is limited by reliance on handcrafted descriptors and reduced robustness under varying imaging conditions.

- **Deep Learning–Based Features**: In contrast to handcrafted features, deep neural networks can learn hierarchical descriptors directly from data. While this reduces manual feature design and improves robustness, it requires large annotated datasets and significant computational resources.

Since not all extracted features are equally relevant, feature selection is often applied to reduce dimensionality and computational load while retaining discriminative power. Popular techniques include Principal Component Analysis (PCA), Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and boosting methods such as AdaBoost [17].

Pre-processing enhances image quality through noise reduction, illumination correction, and segmentation, while feature extraction and selection, transform visual information into descriptors for classification. The effectiveness of these stages directly influences the accuracy of subsequent defect detection and classification.

2.3.3 Detection and Classification Systems

The detection and classification stage constitutes the core of automated visual inspection (AVI) systems, as it determines the ability of the model to correctly identify and categorize weld defects. Approaches range from rule-based methods to traditional machine learning algorithms, deep learning architectures, and hybrid or ensemble strategies, each offering distinct advantages and limitations.

AVI tasks can be broadly categorized into four types [12, 17]:

- **Binary classification:** Distinguishing between defect and non-defect states (e.g., pass/fail), answering the question "is the object in the desired state?"
- **Multi-class classification:** Assigning objects to multiple possible defect states, where outputs span several defect types.
- **Localization:** Identifying the region of the object where a defect occurs.
- **Multi-class localization:** Combining the above two tasks, aligning with segmentation approaches where both the class and location of the defect are determined.

This taxonomy mirrors traditional computer vision tasks of classification, object detection, and segmentation. In practice, weld inspection, often requires **both** classification and localization, since knowing the presence of a crack is insufficient without identifying its position and size.

Figure 2.7 illustrates these three fundamental computer vision tasks.

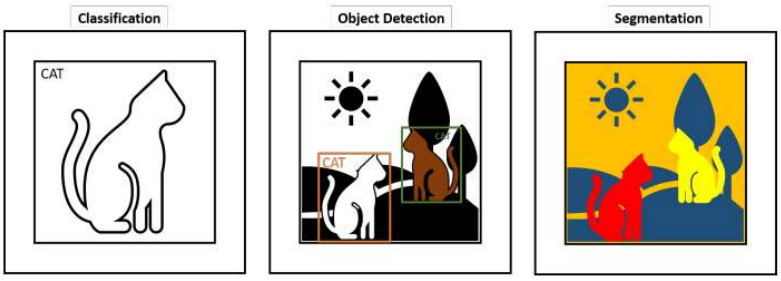


Figure 2.7: Visualization of the three different Computer Vision tasks—classification, object detection with two bounding boxes, and segmentation. [12].

Detection methods can be broadly divided into **rule-based** and **learning-based** strategies. Rule-based approaches, such as thresholding and template matching, are simple, interpretable, and computationally efficient. However, they are highly sensitive to variations in lighting, geometry, or noise, and rely heavily on manual parameter tuning [17]. These limitations motivated the shift toward learning-based classifiers, which can generalize better to complex and variable defects.

Within machine learning, surface defect detection is typically categorized by the learning mode: supervised, unsupervised, or semi-supervised [20].

- **Supervised learning** requires labeled data and is widely adopted in industrial AVI. Neural networks, Naïve Bayes–CNN fusions, and modern object detectors such as Fast R-CNN or Classification Priority Networks (CPN) have achieved high accuracy but suffer from high computational demands and sensitivity to class imbalance.

- **Unsupervised learning** avoids the need for labeled datasets, instead relying on clustering or autoencoders to detect anomalies. These methods are useful when annotations are scarce but are often unstable, sensitive to initialization, and prone to noise.

- **Semi-supervised learning** offers a compromise by leveraging a small number of labeled samples alongside abundant unlabeled data. Generative Adversarial Networks (GANs), residual CNNs, and hybrid CAE–GAN frameworks have demonstrated promising performance, though they typically require many training iterations and careful parameter tuning.

2.3.4 Deep Learning Architectures

While traditional machine learning relies on handcrafted features, deep learning methods learn hierarchical feature representations directly from raw data. This reduces the dependency of manual feature engineering and increases robustness under variable imaging conditions [21]. Convolutional Neural Networks (CNNs), in particular, have achieved state-of-the-art results in tasks such as object detection, image segmentation, and classification, making them highly suitable for weld defect inspection.

Convolutional Neural Networks (CNNs)

CNNs are composed of convolutional layers that extract spatial features, pooling layers that reduce dimensionality, and fully connected layers for decision making. Modern architectures such as AlexNet, VGGNet, ResNet [22], and Inception have been widely adopted in industrial inspection. Figure 2.8 illustrates their structure.

Key features include:

- **Strengths:** CNNs automatically learn multi-level features (from edges and textures to object-level patterns), are translation-invariant, and scale effectively to large datasets.

- **Weaknesses:** They require substantial amounts of labeled data, are computationally intensive, and their performance can degrade in highly imbalanced datasets typical of defect inspection.

Object Detection Networks

Beyond classification, weld inspection often requires localizing defects. CNN-based detectors are generally categorized into two families:

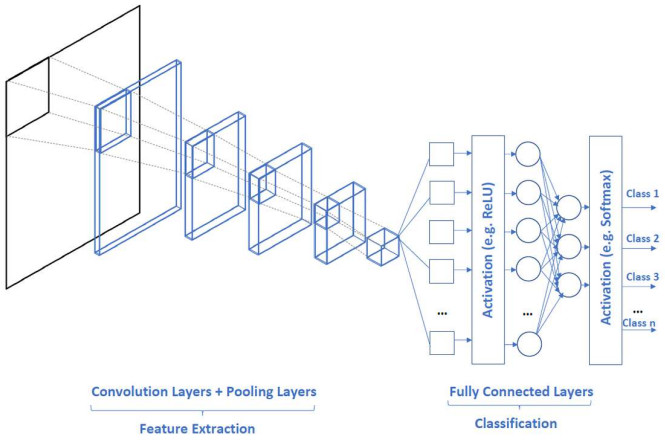


Figure 2.8: CNN structure [17].

- **Two-stage detectors:** such as R-CNN, Fast R-CNN, and Faster R-CNN, which first generate region proposals and then classify them. These methods achieve high accuracy but are computationally expensive.
- **Single-stage detectors:** such as YOLO (You Only Look Once) [23] and SSD (Single Shot Multibox Detector), which directly predict bounding boxes and classes in a single pass. They are significantly faster but may trade accuracy for speed, especially in small-defect detection.

Figure 2.9 illustrates the YOLO model detection as a regression problem.

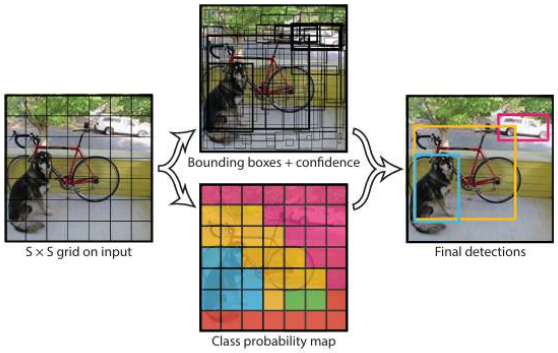


Figure 2.9: YOLO model [23].

Transformer-Based Architectures

Transformer models, initially developed for natural language processing, have recently shown strong potential in computer vision by leveraging self-attention mechanisms to capture long-range dependencies. Several architectures are particularly relevant for AVI:

- **Vision Transformer (ViT):** Introduces self-attention for image recognition, effectively modeling global context. However, it struggles with high-resolution tasks and

requires very large datasets to generalize well [24]. Figure 2.10 illustrates the model overview of a ViT model.

- **Swin Transformer:** Improves efficiency and scalability through hierarchical design and shifted windows, achieving competitive results in classification, detection, and segmentation [25].
- **Dense Prediction Transformer (DPT):** Specializes in dense prediction tasks such as segmentation, making it valuable for precise defect localization.
- **ConvNeXt:** Modernizes ResNet-like CNNs by incorporating design principles from Transformers, achieving comparable or superior performance while retaining convolutional efficiency [26].

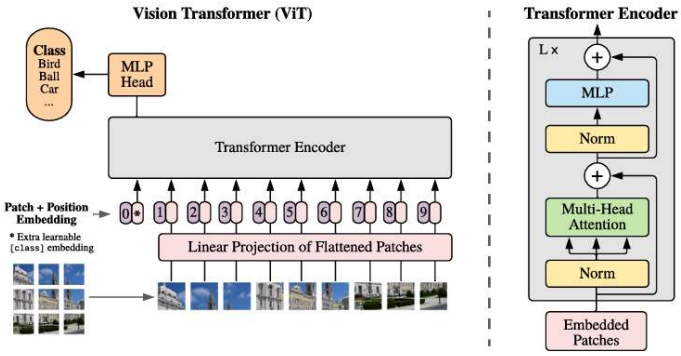


Figure 2.10: ViT model overview [24].

Transformer-based models excel at capturing global context, adapt flexibly across tasks, and have set benchmarks in classification and dense prediction. However, they demand substantial computational resources and large-scale datasets, which can limit their applicability in data-scarce industrial environments.

Semantic Segmentation Networks

When precise defect localization and shape analysis are required, segmentation architectures are preferred.

- **Fully Convolutional Networks (FCNs)** replace fully connected layers with convolutional layers to produce pixel-level predictions.
- **U-Net and its variants** introduce skip connections between encoder and decoder stages, enabling accurate segmentation with limited training data, which has made them highly popular in industrial and medical imaging.

Segmentation networks are sensitive to annotation quality and typically demand more computation and memory than classifiers.

Transfer Learning and Data Augmentation

In AVI applications, the scarcity of labeled defect images is a major limitation. Transfer learning mitigates this by fine-tuning networks pre-trained on large datasets such as ImageNet. Data augmentation (e.g., rotations, flips, scaling, noise injection) further improves generalization by synthetically expanding training datasets, although large geometric changes can be harmful in certain cases, such as weld inspection [9].

Hybrid and Ensemble Methods

Recent research has explored combining CNNs with traditional machine learning classifiers or fusing multiple deep learning architectures. Examples include CNN–SVM hybrids, ensemble voting systems, and GAN-based data generation to address imbalance. While these approaches can boost accuracy and robustness, they also increase system complexity and training cost.

Limitations

The deployment of deep learning based AVI systems for weld defect detection requires balancing accuracy, efficiency, and interpretability. While the architectures reviewed above provide strong foundations, several practical factors and limitations shape their real-world adoption:

- **Performance:** Models must achieve accuracy comparable to or exceeding human inspectors, with error priorities (false positives vs. false negatives) depending on the industrial process.
- **Real-Time Capability:** Applications demand high throughput, typically measured in frames per second (FPS) [12, 27].
- **Hardware Constraints:** Edge deployment requires lightweight architectures and optimized inference pipelines [12].
- **Data Scarcity:** Labeled defect datasets are limited, motivating transfer learning, augmentation, and synthetic data generation [28].
- **Generalization Ability:** Models must remain robust under variations in lighting, positioning, and defect morphology.
- **Explainability and Trust:** Interpretability is essential for adoption in safety-critical industrial settings [12].

Overall, while deep learning delivers state-of-the-art performance in weld defect inspection, overcoming these challenges is crucial for industrial-scale deployment. Research directions such as lightweight CNNs, efficient convolutional operations, semi-supervised

learning, advanced augmentation strategies, and domain adaptation are especially promising. Figure 2.11 summarizes representative AI algorithms explored for NDT of welding defects, including machine learning, ensemble, and deep learning approaches.

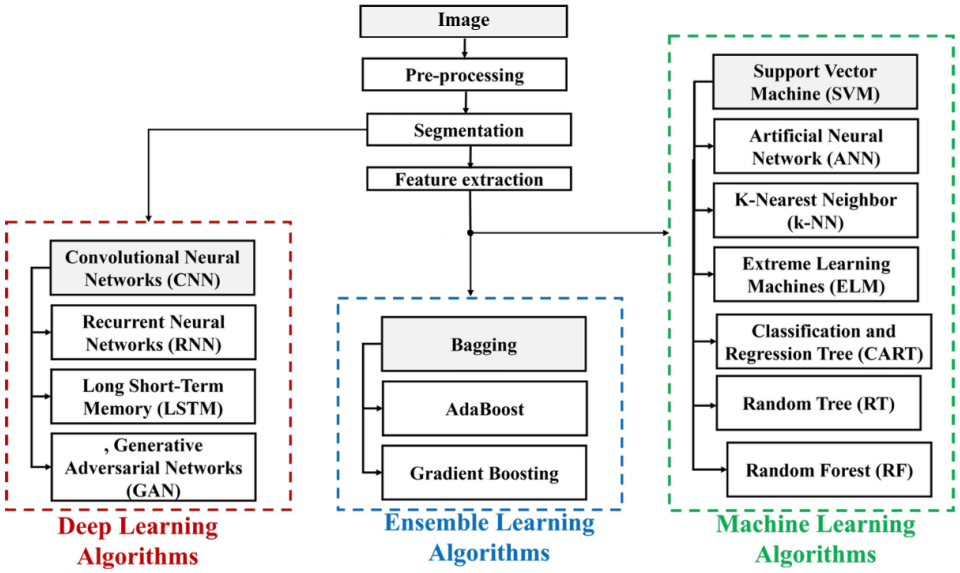


Figure 2.11: Representative AI algorithms for NDT of welding defects, spanning machine learning, ensemble learning, and deep learning (adapted from [8].)

2.3.5 Data Challenges

A major barrier to robust AVI deployment is the scarcity and imbalance of annotated datasets. Many industrial studies rely on relatively small collections (median ~2000 images), often with highly skewed class distributions [12]. This imbalance leads to overfitting and poor generalization, particularly when defect classes are underrepresented. Deep learning methods are thus strongly dependent on high-quality data, yet industrial environments pose challenges including sparsity, imbalance, and domain shifts.

Table 2.1 summarizes the most common data quality issues in deep learning based defect detection, ranging from practical difficulties in collecting sufficient annotated data to environmental and sensor-related problems that degrade input quality [28].

To mitigate these issues, several complementary strategies are employed:

- **Classical Data Augmentation:** Geometric transformations (flipping, rotation, scaling, cropping), intensity adjustments (contrast, brightness, color jittering), kernel filtering, mixing images, and random erasing. These techniques reduce overfitting and improve robustness, though their effectiveness depends strongly on dataset characteristics [28].

- **Advanced Data Augmentation:** Complex transformations such as Gaussian noise, perspective warping, or Moving Least Squares have shown superior improvements compared to simple augmentations, particularly in noisy industrial conditions [28].

Table 2.1: Causes of data quality issues in deep learning based visual defect detection [28].

Data Quality Issue	Description
Amount of data	Difficulty in collecting sufficiently large amounts of labeled samples.
Label inconsistencies	Annotation is labor-intensive, ambiguous, and often requires multiple experts.
Data imbalance	Defective parts are underrepresented compared to non-defective samples.
Changing lighting conditions	Contrast and brightness variations across work shifts.
Exposure issues	Reflections and shadows caused by complex component geometry.
Sensor failure	High noise levels or degraded images due to sensor malfunction.
Changing object poses	Different orientations of components in mass production.
Changing appearances	Product appearance changes over time, making earlier data unusable.

- **Synthetic Data Generation:** CAD models, rendering software, and domain randomization pipelines enable the large-scale creation of training datasets at low cost. Hybrid datasets (e.g., 90–95% synthetic, 5–10% real data) consistently outperform purely real or purely synthetic ones, effectively alleviating annotation bottlenecks while maintaining real-world accuracy [16]. Figure 2.12 illustrates a synthetic data generation pipeline compared to traditional processes.

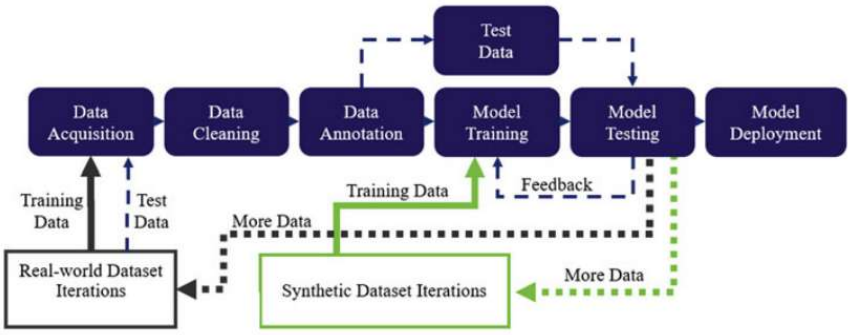


Figure 2.12: Synthetic training data generation pipeline compared to traditional real data pipeline (adapted from [16]).

- **Regularization and Sampling:** Dropout, L2 regularization, cost-sensitive learning, and under-/over-sampling strategies address imbalance and reduce overfitting [17].

- **Transfer Learning and Semi-supervised Learning:** Fine-tuning pre-trained networks and employing methods such as MixMatch leverage large external datasets and unlabeled samples to improve generalization in data-scarce domains [6, 15].

2.4 State of the Art in Automated Visual Inspection Systems

The growing demand for accuracy, speed, and cost-effectiveness in industrial environments has accelerated the adoption of AVI systems. These systems combine advanced imaging technologies with deep learning algorithms for real-time defect detection and classification.

2.4.1 Advanced Imaging Techniques

One of the primary challenges in weld inspection is obtaining high-quality images under varied conditions, such as fluctuating light levels and surface reflectivity. Nguyen et al. (2011) [29] addressed this with a laser vision sensor system based on laser triangulation, illustrated in Figure 2.13.

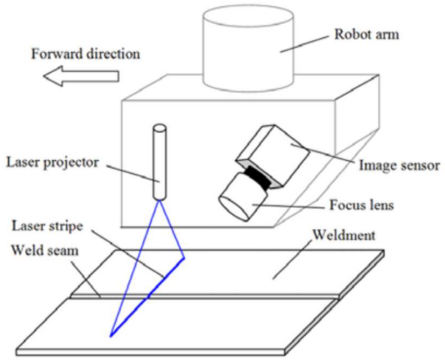


Figure 2.13: Laser vision sensor module using laser triangulation for 3D weld profiling [29].

The system projects a laser stripe onto the weld surface, and a camera captures the reflected light to reconstruct a 3D weld profile. High-speed GigE cameras enable acquisition at up to 135 frames per second, supporting real-time inspection [29].

Sikström et al. (2025) [30] explored optical non-contact 3D scanning for weld distortion measurement. By integrating robotics and Finite Element Analysis (FEA), their approach reduces geometrical distortions while highlighting challenges from specular reflections on highly reflective metals.

Robotic integration with advanced vision systems has further enhanced AVI capabilities. Ivanov et al. (2022) [31] implemented the FANUC iRVision 3DL system, combining 3D laser sensors with cameras to detect surface defects like pores, cracks, and undercuts during welding (Figure 2.14).

For underwater inspections, Baldez et al. (2025) [32] used remotely operated vehicles (ROVs) equipped with imaging and non-destructive evaluation tools. Figure 2.15 illustrates original images at varying turbidity and corresponding texture-verified defect regions.

2.4.2 Automated Defect Classification

Standardizing defect classification in AVI has been an ongoing research focus. Nizam et al. (2025) [33] reviewed various approaches, emphasizing the challenge of detecting

Figure 2.14: Integration of the FANUC iRVision 3DL system into a welding robot for surface defect detection [31].

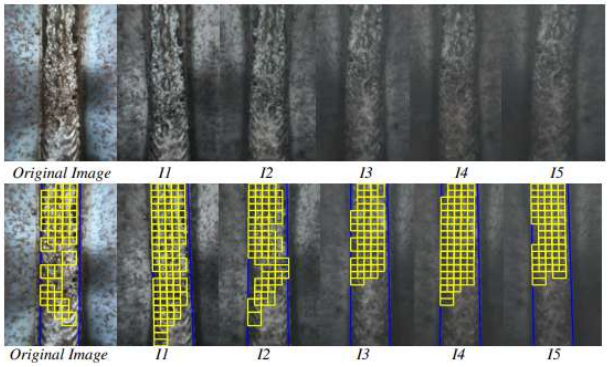


Figure 2.15: Underwater weld inspection: original images under varying turbidity and texture-based inspection highlighting potential irregularities [32].

small flaws. They explored grayscale profile-based systems and geometrical parameter extraction to improve feature representation without requiring full segmentation.

Deep learning, particularly CNN-based architectures, has significantly improved weld defect classification compared to traditional template-matching methods. There is improvement where the integration of Mask R-CNN with Slicing Aided Hyper Inference enables accurate segmentation and detection of weld defects, effectively capturing small flaw regions that are often challenging for conventional approaches.

Lightweight CNNs (MobileNet, ShuffleNet, SqueezeNet) also allow faster inference with minimal accuracy loss [12]. Transfer learning, semi-supervised approaches, and hybrid models — combining deep learning with traditional techniques — enhance performance on smaller or partially labeled datasets [6].

Transformers have recently set new benchmarks in weld defect detection, though, as mentioned before, they require large datasets and resources. To mitigate these limitations, Liu et al. (2022) [34] proposed the Fast Multi-Path Vision Transformer (FMPVit), which combines transformer-based global context with convolutional modules for local detail.

Compared to CNNs and mainstream transformers, FMPVit achieved higher accuracy with fewer parameters, offering a more efficient balance between performance and complexity.

The practical impact of these approaches is evident across multiple industries, where AVI systems are tailored to domain-specific requirements [19]:

- **Electronics:** ResNet50 and GoogLeNet classify semiconductor and wafer defects via image segmentation.
- **Automotive:** AI-powered vision detects scratches and verifies compliance with quality standards.
- **Ceramics:** Fast R-CNN and YOLO improve detection of surface defects.
- **Aerospace:** DNNs and CNN–SVM hybrids identify structural damage in aircraft components.
- **Additive Manufacturing:** Real-time defect detection during powder bed fusion is enabled by AI-driven systems.
- **Textile and Metallurgy:** Deep CNNs and Transfer Learning-based U-Net (TLU-Net) identify fabric and metal surface flaws.

2.4.3 Challenges and Future Directions

Despite rapid progress, AVI systems still face challenges such as limited and imbalanced datasets, high computational demands, difficulties with lighting and material reflectivity, limited explainability of deep models, and integration into industrial workflows.

Future AVI systems are expected to integrate smart cameras capable of real-time image processing, with CMOS sensors dominating due to built-in processing functions that boost efficiency and reduce costs [20]. Embedded AI in smart cameras will further enhance real-time defect detection, enabling edge computing solutions that reduce latency and data transfer demands.

Future research is also expected to advance the field by developing lightweight and hybrid architectures, leveraging semi-supervised and transfer learning, and applying advanced data augmentation.

3

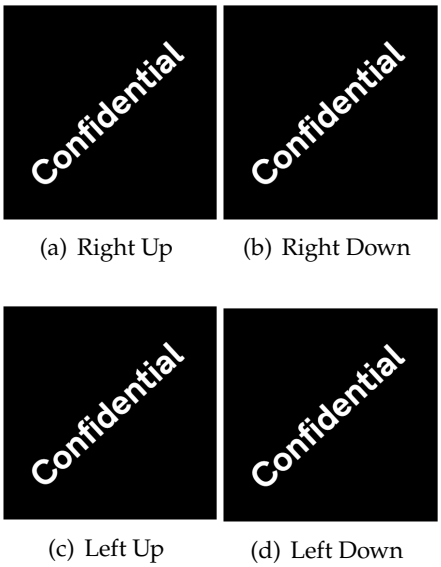
Experimental Procedure

3.1 Test Component

The image acquisition system was designed to capture welds on the test component shown in Figure 3.1. The component has four primary welds connecting the main parts and three additional welds joining subcomponents. For this study, only welds 1, 2, and 3 are considered, referred to as Zone 1, Zone 2, and Zone 3 throughout this dissertation.

Figure 3.1: Welds on the test component.

For each component, images were collected from six positions, referred to as Top 1, Top 2, Top 3, Bottom 1, Bottom 2, and Bottom 3. Components were distinguished as left or right pieces. Figures 3.2(a) and 3.2(d) illustrate the top and bottom positions for both right and left components.



(a) Right Up (b) Right Down

(c) Left Up (d) Left Down

Figure 3.2: Top and bottom zones on the right and left components.

3.2 Experimental Apparatus

This section details the equipment used for the image acquisition. The assembled setup is shown in Figure 3.3, including the test component, support structure, image acquisition system (with four possible positions), top screen, and background screen. The top screen minimizes ambient light interference, while the background screen enhances contrast between the component and surroundings.

Figure 3.3: Support structure and image acquisition system.

3.2.1 Support Structure

A dedicated support structure was designed to ensure reproducibility in component positioning. Key objectives included:

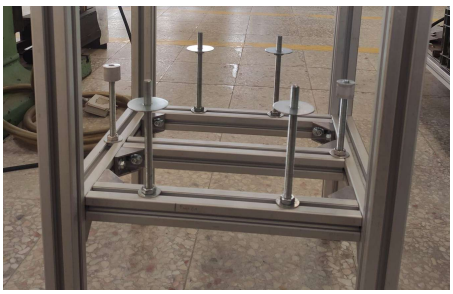
- Elevate the component above the ground.

- Facilitate efficient replacement of components.
- Maintain consistent positioning of components for accurate imaging.
- Provide stable mounting points for the acquisition system.

The component was supported at three points: two via threaded shafts, nuts, and washers in the openings of the component, and one near Zone 2 using a custom support. The structure allowed top and bottom component orientations, corresponding to the welds being imaged. Figures 3.4(a), 3.4(b), and 3.4(c) illustrate the support structure and component positioning.

(a) Top component position

(b) Bottom component position



(c) Detailed view of component supports

Figure 3.4: Support structure and component positioning.

Four camera positions were designated for the acquisition system above the support to capture all required top and bottom views of Zones 1, 2, and 3.

3.2.2 Image Acquisition System

The image acquisition system consisted of:

- Deheng Mer2-2000-6GM monochrome camera (20 MP)
- Compulab V1624-MPZ lens (16 mm f/2.5) with polarizer
- EFFILUX ring lighting system with diffuser and polarizer

- Custom acquisition software developed for this purpose

Figure 3.5 shows the assembled system. Four positions were required to capture all six weld views. Zones 1 and 3 could be imaged from a single position by flipping the component, while Zone 2 required separate top and bottom positions.

Figure 3.5: Image acquisition system assembly.

Prior to image acquisition, the system was calibrated for camera alignment, and the positions were fixed to prevent camera movement during imaging.

3.3 Methodology

In this study, two image acquisition tests were performed, with only right-hand components being captured.

First Test

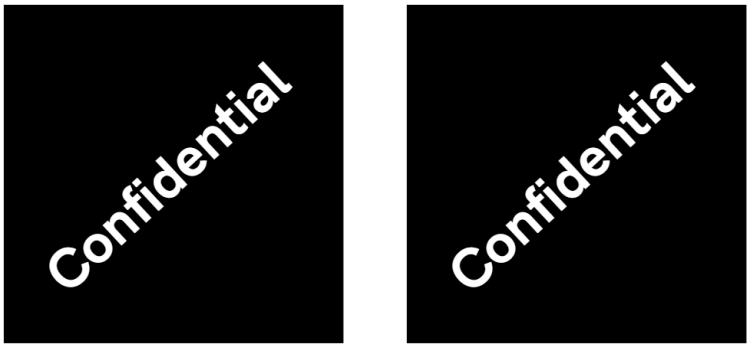
The first test included 53 components and a total of 318 images were captured, with three consecutive images per position. Several issues were identified:

- Post-production defects (scratches, impact marks, ink annotations, rust) interfering with defect detection training.
- Slight component movement in the support reducing reproducibility.
- Difficulty in positioning components efficiently.
- Low edge contrast due to textured background, hindering alignment.

Second Test

To address the issues found on the first test, improvements implemented for the second test included:

- Components were imaged immediately after production to minimize handling and waiting time.
- A white background was placed beneath each component to enhance edge contrast. Figures 3.6(a) and 3.6(b) illustrate the difference in image quality with and without the white background.



(a) With white background (b) Without white background

Figure 3.6: Comparison of image quality with and without white background.

For the second test, images were captured for Zone 2 only: 41 components at Top 2 and 33 components at Bottom 2, with three consecutive images per position, totaling 222 images.

Defect Labeling

Each image from both tests was labeled with one of the following categories: None, Lack of Fusion, Welding Overlay, Welding Reduction, Spatter, Porosity, Bead Height, Other Defects, or Missing Component.

4

Dataset

The creation of the dataset required for the defect recognition system involved several decisions aimed at facilitating later stages of this work.

In the first test, images were captured from 53 components, with three consecutive images per position, totaling 318 unique images. Considering the consecutive images, the total number of images amounts to 954.

The second test captured only two positions per component: Top 2 and Bottom 2. For this test, 41 components were imaged at Top 2 and 33 at Bottom 2, with three consecutive images per position, totaling 222 images.

Tables A.1 and A.2 summarize the total number of images per defect and their distribution in the first and second tests, respectively. These tables are provided in Appendix A.

Following a careful review, some images, including the ones initially labeled as *Other Defects*, were reassigned to the correct defect categories. The updated dataset is summarized in Table 4.1.

Table 4.1: Updated total number of images per defect per position

Defect/Position	Bottom 1	Top 1	Bottom 2	Top 2	Bottom 3	Top 3	Total
None	132	150	238	196	159	159	1034
Lack of Fusion	3	9	9	3	0	0	24
Porosity	0	0	0	3	0	0	3
Bead Height	24	0	0	0	0	0	24
Weld Overlay	0	0	21	6	0	0	27
Total	159	159	268	208	159	159	1112

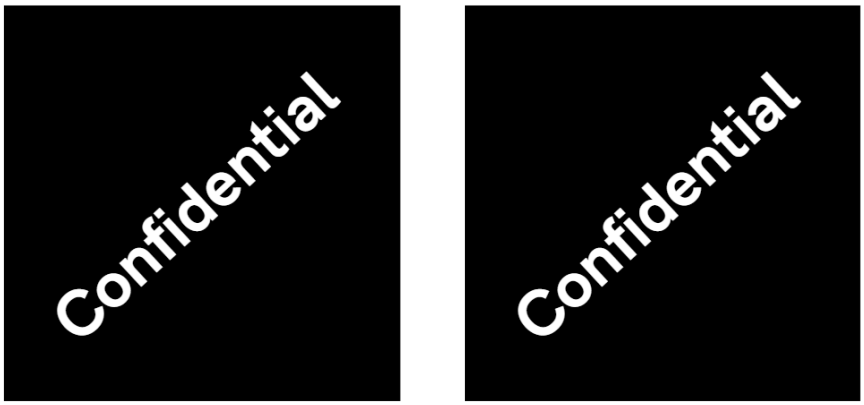
4.1 Image Preprocessing

The image preprocessing, conducted by another researcher, addressed some of the issues found during image acquisition:

- Slight movement of components within the support, causing non-reproducible placements.

- Post-production markings, such as ink annotations.

Some of those markings clearly identified the defect, which could mislead the model into learning the annotations as defect features rather than the defects themselves, examples of this are illustrated in Figure 4.1.



(a) Image with annotations (b) Image with ink/impact markings

Figure 4.1: Examples of images containing markings and annotations.

4.1.1 Alignment

To correct misalignments, caused by the slight movement of the components, all images were aligned to a reference image (which typically is the first acquired). The edges of the component were manually identified in the reference image, and a correction matrix was computed to align all other images in the XY plane, including rotation, but excluding geometric distortions.

The alignment was easier in the second test due to the uniform white background, which improved edge detection, particularly in positions like Top 3, which have limited reference points, as illustrated in Figure 4.2.

Figure 4.2: Edge identification of the component for alignment; primarily referenced along one axis.

4.1.2 Region of Interest Mask

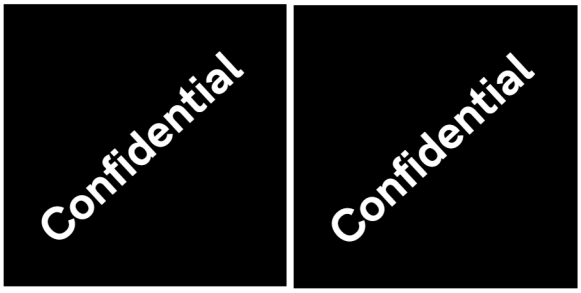
After alignment, the Region of Interest (ROI) was isolated using the following steps (Figure 4.3):

(a) Original aligned image.

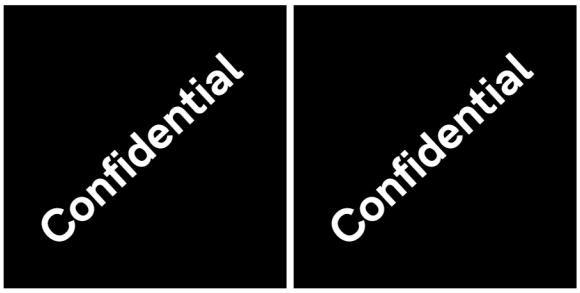
(b) Gamma correction applied ($\gamma = 1.5$) to enhance darker tones; no additional brightness or contrast adjustments were made.

(c) Binary mask applied, manually drawn using an image editing program.

(d) Cropping of the image to the ROI defined by the mask, producing a final resolution of 2024×2024 pixels.



(a) Original Image (b) Gamma Correction



(c) Application of the mask (d) Crop to ROI

Figure 4.3: Preprocessing steps applied to each image.

This preprocessing reduces noise, removes annotations, and standardizes images to help models learn defect features rather than context cues.

4.2 Exploratory Data Analysis

A dataframe was created containing the metadata for each image, including date, time, component side, zone, defect type, filename, image number, path, relative path, component ID, and optional comments.

No null values or anomalies were found. However, a severe class imbalance was identified, as illustrated in Figure 4.4(a) and Table 4.2. Additionally, the distribution of defects across zones was found to be non-uniform, as shown in Figure 4.4(b) and Table 4.3.

Table 4.2: Number of images per defect type

Label	Image Count
None	1024
Lack of Fusion	24
Porosity	3
Bead Height	24
Weld Overlay	27
Total Number of No Defects / Defects	1024 / 78

Table 4.3: Percentage of defects per zone

Position	Defect	No Defect	Total	% Defects
Bottom 1	27	132	159	17.0%
Bottom 2	30	228	258	11.6%
Bottom 3	0	159	159	0.0%
Top 1	9	150	159	5.7%
Top 2	12	196	208	5.8%
Top 3	0	159	159	0.0%
Total	78	1024	1102	7.1%

In summary, several key concerns were identified. Most models require at least 50–100 images per defect class to generalize effectively, yet the dataset contains a maximum of 27 images per defect and as few as 3 images for some classes.

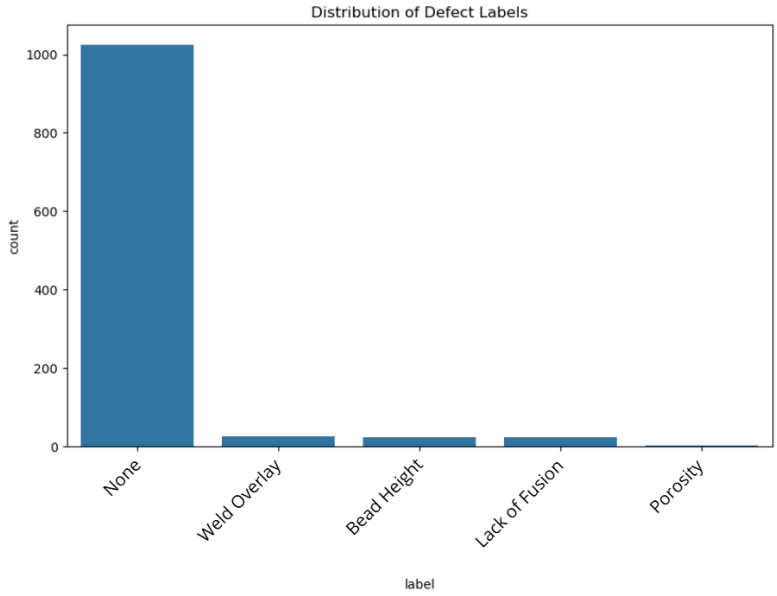
Only one-third of the defect images are unique, with the remainder being duplicates, which increases the risk of model overfitting. The Porosity class includes just a single unique image, making it unsuitable for validation under strict no-leakage protocols, as it will be explained later.

Furthermore, Zone 3 contains no defect images, preventing validation in that region. These findings highlight a major limitation for model development.

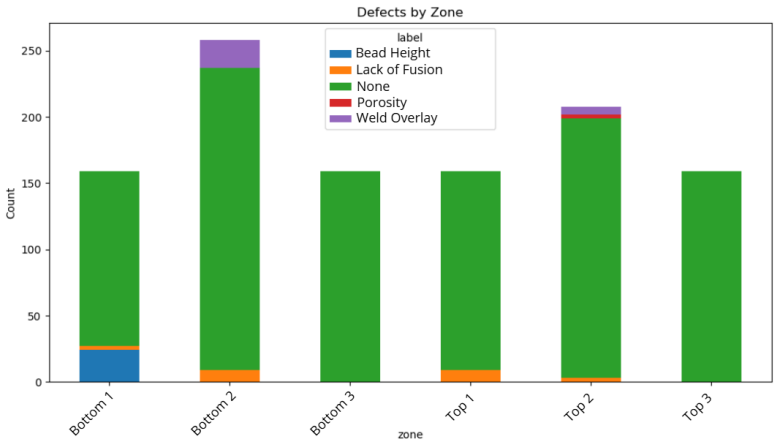
4.3 Dataset Preparation

The Dataset was split into training and validation sets, and a held-out test set. The test set represents 20% of the total dataset. Validation was used for model selection and hyperparameter tuning, while final performance was reported on the held-out test set. Oversampling and augmentation were applied only to the training set to prevent data leakage.

The distribution of the test dataset is shown in Table 4.4.



(a) Class distribution of defects in the dataset



(b) Distribution of defects by zone

Figure 4.4: Comparison of defect distributions in the dataset.

Table 4.4: Distribution of classes in the Test Dataset

Label	Image Count
None	73
Lack of Fusion	3
Porosity	0
Bead Height	3
Weld Overlay	6
Total Number of Images	85

5

Model Development

5.1 Architecture of the Detection and Classification System

As stated in Chapter 1, the main objective of this work is the development of an automatic detection and classification system for surface defects on welds using computer vision. The complete system integrates both the hardware described in Section 3.2 and a software pipeline that combines defect detection and classification using fine-tuned pre-trained models.

Three pipeline configurations were considered:

1. **Global Pipeline**: A single pipeline that performs detection followed by classification using a single unified model ensemble.

2. **Classification Pipeline**: A configuration focusing primarily on the classification stage.

3. **Hybrid Pipeline**: A combined approach that integrates the strengths of both pipelines and multiple models, designed to maximize both detection and classification accuracy, especially for minority defect classes.

The developed pipelines enables the system to automatically process weld images and output predictions with high confidence.

The overall architecture of the detection and classification pipeline (global pipeline) is illustrated in Figure 5.1.

The workflow begins with weld image acquisition. Images are pre-processed as described in Section 4.1, then passed through the detection–classification pipeline.

The detection stage is handled by a YOLOv8 model [23] trained with binary labels (*defect* vs. *no defect*). If no defect is detected, the pipeline terminates, since defect-free welds require no further processing. If a defect is detected, the model outputs a bounding box (bbox) around it, which is used to crop the defective region.

Cropped images are subsequently passed to a classification model fine-tuned on images of cropped welded regions containing both defects and non-defects. This model assigns a

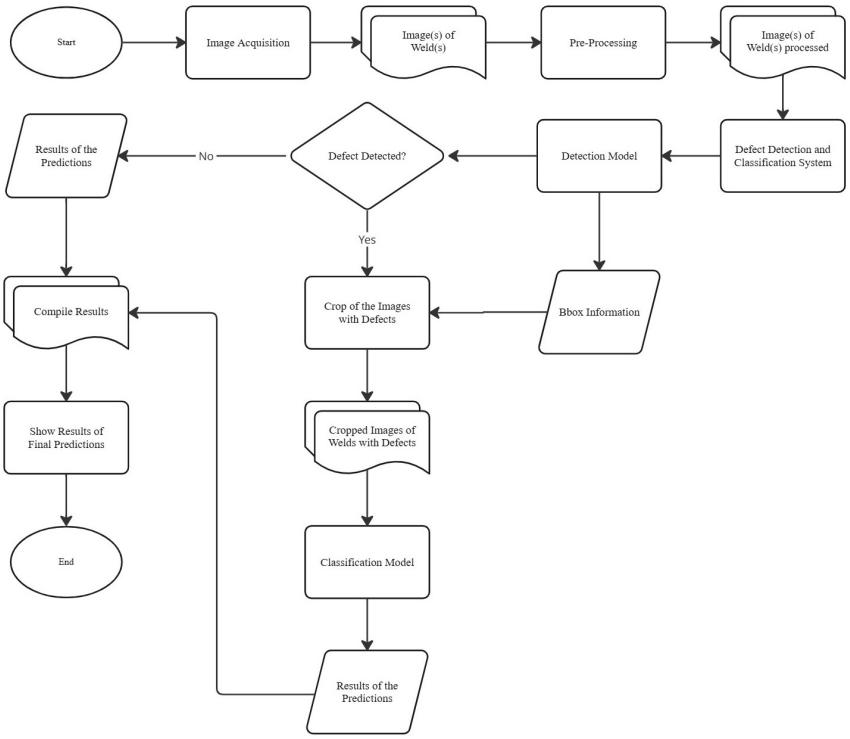


Figure 5.1: Global pipeline for weld defect detection and classification.

specific defect type (e.g., lack of welding, porosity, weld overlap, or excessive/insufficient bead height). Finally, the predictions from detection and classification are compiled as the system output.

This stepwise logic ensures that classification is only performed when necessary, thereby reducing computational cost and minimizing false positives caused by sending intact welds to the classifier.

For comparison, a baseline architecture was also implemented (Figure 5.2). In this simpler pipeline, classification is applied directly to full weld images without a prior detection stage. While computationally simpler, this approach ignores localized information and generally performs worse than the detection–classification pipeline.

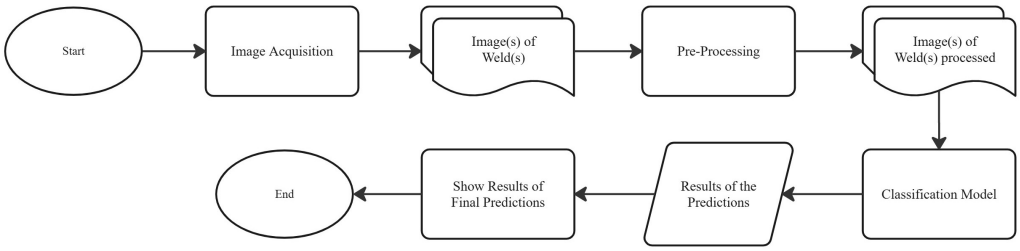


Figure 5.2: Baseline pipeline applying classification directly to full weld images.

A hybrid pipeline combining both approaches was also implemented. In this design, if the detection model failed to identify a defect (and thus no crop was produced), the

full-image classifier acted as a fallback to recover potential misdetections. The hybrid architecture is shown in Figure 5.3.

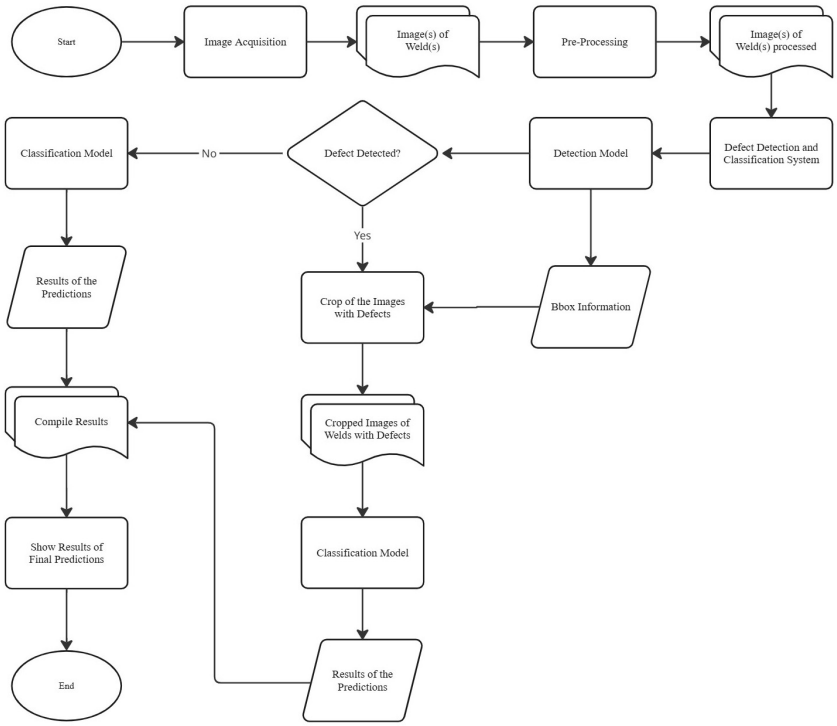


Figure 5.3: Hybrid pipeline combining detection-first with fallback classification.

5.2 Evaluation Strategies and Metrics

The proposed models were evaluated using standard machine learning practices, adapted to the specific challenges of this work such as the severe class imbalance and multiple defect categories.

Evaluation metrics are presented within the context of the detection and classification tasks, with formal definitions included to clarify how performance was quantified.

For both both tasks, predictions can be summarized as:

- **True Positives (TP)**: correctly predicted defects, or correctly detected defects with the correct class and sufficient bounding-box overlap,
- **False Positives (FP)**: predictions of a defect where none exists, detections with no ground-truth match, or detections assigned to the wrong class,
- **False Negatives (FN)**: missed detections or misclassified defects.

5.2.1 Metrics for Detection

Detection performance was measured using the *mean Average Precision* (mAP), a standard metric in object detection that accounts for both localization and classification accuracy.

Precision and Recall quantify the ability to correctly detect defects while avoiding false alarms. At a given Intersection over Union (IoU) threshold, predictions are correct if the predicted bounding box overlaps the ground truth by at least τ (with $\tau = 0.5$ in this work) and the class is correct:

$$\text{Precision} = \frac{TP}{TP + FP}, \tag{5.1}$$

$$\text{Recall} = \frac{TP}{TP + FN}. \tag{5.2}$$

High Precision indicates few false positives, while high Recall indicates few missed defects. Here, for the detection task, True Positives require both correct classification and sufficient bounding-box overlap.

Precision–Recall (PR) Curve: Since Precision and Recall vary with the detection confidence threshold, a PR curve is used to summarize this trade-off. The area under this curve corresponds to the Average Precision.

Average Precision (AP) and Mean Average Precision (mAP)

The Average Precision (AP) for a defect class is the area under the PR curve, which is obtained by varying the detection confidence threshold:

$$AP_c = \int_0^1 \text{Precision}_c(\text{Recall})\, d(\text{Recall}). \tag{5.3}$$

The mean Average Precision (mAP) is then computed as the mean of AP values across all classes:

$$mAP = \frac{1}{C} \sum_{c=1}^{C} AP_c, \tag{5.4}$$

where C is the number of classes.

Following COCO conventions [35], results are reported as:

- mAP_{50}: mAP at IoU threshold 0.5,
- mAP_{50-95}: mAP averaged across IoU thresholds from 0.5 to 0.95 in steps of 0.05.

5.2.2 Metrics for Classification

Classification models were evaluated using both overall and class-specific metrics, with particular attention to class imbalance.

Accuracy is defined as:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}. \tag{5.5}$$

Precision, Recall, and F1-Score: For each class c, Precision and Recall are as in Equations 5.1 and 5.2. The F1-score balances Precision and Recall:

$$F1_c = 2 \cdot \frac{\text{Precision}_c \cdot \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c}. \tag{5.6}$$

Macro vs. Weighted Averages: In multi-class classification, results are aggregated as:

- **Macro-F1**: unweighted mean of F1-scores across classes, treating each equally,
- **Weighted-F1**: mean weighted by class support (number of samples).

Macro-F1 emphasizes minority classes, while Weighted-F1 reflects performance under real-world class distributions. In this work, Macro-F1 was chosen as the primary metric to evaluate and compare classification models, as it provides the fairest assessment across all defect classes.

Confusion Matrix: Confusion matrices visualize class-specific performance. Diagonal entries indicate correct classifications (TP), while off-diagonals correspond to misclassifications (FP or FN).

K-Fold Cross-Validation: Classification models were trained and evaluated with k-fold cross-validation. Metrics were averaged across folds to estimate generalization and avoid overestimating performance from a single split. During training, the following were tracked:

- Training and validation loss,
- Training and validation Macro-F1,
- Per-class validation F1-scores.

After training, ensemble models were built from the fold models. These were evaluated with Accuracy, Macro-F1, Weighted-F1, and per-class metrics. While ensembles often achieve higher performance, they can be overly optimistic since they are trained on the full dataset. Thus, fold-averaged results (e.g., mean Macro-F1 across folds) are considered the more reliable indicator of generalization and were used for the evaluation of the classification models in this work.

5.2.3 Pipeline Evaluation

The integrated detection–classification pipeline was also evaluated as a whole, since performance depends on the interaction between detection and classification.

Threshold Sweep To select the optimal YOLO confidence threshold, values from 0.1 to 0.9 were tested. For each threshold, the following were recorded:

- Pipeline accuracy,
- Macro-F1 of the classification output,
- Number of images passed from YOLO to the classifier.

Macro-F1 was chosen as the selection criterion since it balances class performance and mitigates dominance of the majority class. The threshold maximizing Macro-F1 was used in the final pipeline.

Fallback Classification In the hybrid architecture, fallback full-image classification was triggered when YOLO failed to detect defects. Metrics from these predictions were integrated with cropped-image results, providing a complete evaluation.

In summary, the evaluation strategies presented in this section are tailored to each specific task: Defect detection is assessed primarily using mAP across IoU thresholds, supported by PR curves to analyze Precision–Recall trade-offs; Defect classification is evaluated using mean fold-level Macro-F1 and per-class F1-scores, complemented by ensemble-level Accuracy, Macro-F1, Weighted-F1, and confusion matrices. By combining these task-specific metrics with pipeline-level evaluations, including threshold sweeps and fallback classification, the methodology provides a fair and realistic assessment of model performance under severe class imbalance and varied defect visibility.

5.3 Data Augmentation

Data augmentation was a central strategy to address the limited dataset size and severe class imbalance identified in Section 4.2.

Since deep learning models typically require thousands of samples per class to generalize effectively, augmentation was employed to increase variability, improve robustness, and reduce overfitting. The augmentation pipeline evolved through several stages, reflecting methodological refinements and insights from preliminary experiments:

Stage 1: Baseline Augmentations Initial experiments used standard geometric transformations common in image classification: resizing images to 224×224, random horizontal flips, and small random rotations ($\pm 10°$). Pixel intensity normalization (mean and standard deviation = 0.5) was applied. Validation and test sets were only resized and normalized.

Stage 2: Enhanced Geometric Distortions In this stage, additional transformations were introduced in the training set to improve model generalization. Random affine transformations were applied with scaling factors in the range $[0.95, 1.05]$, together with

random perspective distortions using a factor of 0.2. Standard augmentations such as flips and rotations were also maintained to further increase variability in the dataset.

Stage 3: Class-Specific Augmentation To address the class imbalance, where the *None* class dominated the dataset, augmentations were applied asymmetrically across classes.

The Majority class (*None*) was subjected only to mild transformations such as horizontal and vertical flips, while minority classes received stronger augmentations. These included affine transformations, color jittering, perspective distortions, flips, and rotations, ensuring greater diversity in the underrepresented categories.

A custom `Dataset` class applied augmentations depending on class membership, enlarging minority-class variability.

Stage 4: MixUp Regularization MixUp was introduced to mitigate overfitting, generating synthetic examples by linearly interpolating two images and their labels. A parameter $\alpha = 0.4$ controlled interpolation strength. The adapted loss combined contributions from both labels. MixUp was first applied uniformly, then tested only on minority classes.

5.4 Defect Detection Model

The defect detection task was addressed using the YOLOv8 object detection framework [23]. Given the limited size and imbalance of the dataset, a lightweight YOLOv8n backbone was selected. This variant provides a good trade-off between speed and accuracy while remaining feasible to train with limited GPU resources.

All training and inference experiments were conducted on Google Colab, leveraging GPU acceleration. This avoided the prohibitively long runtimes that would have been required on a local CPU-only machine.

Training Setup

The dataset, as previously mentioned, was split into training (80%) and validation (20%) subsets, resulting in 882 training and 220 validation images. Because the number of defect samples was insufficient to train a detection model effectively, defect images in the training set were oversampled by a factor of five. This resulted in a more balanced training set of 1142 images.

The model was trained for up to 100 epochs with early stopping (patience = 30), using a batch size of 16 and an input resolution of 640×640.

Training incorporated YOLOv8's built-in augmentation strategies, including random rotation, translation, scaling, perspective distortion, flipping, HSV-based color jitter, Mosaic, and MixUp. In later experiments, Copy–Paste augmentation was introduced to artificially enrich positive samples by pasting cropped defects into background regions of other images.

Two detection strategies were explored:

- **Multi-class detection:** each defect type treated as a separate class.
- **Binary detection:** all defect types collapsed into a single "Defect" class versus "No Defect."

Binary detection was ultimately preferred for the integration in the pipeline. The limited number of examples per class hindered stable multi-class training, whereas binary detection provided a more reliable foundation. In this setup, the detector served primarily as a localization stage, leaving fine-grained defect classification to a downstream model trained on cropped regions.

Outputs

Predicted bounding boxes were visually inspected against ground-truth annotations to verify consistency. An example is shown in Figure 5.4, where ground-truth boxes are in red and YOLOv8n predictions in green.

Figure 5.4: Comparison of ground-truth (red) and predicted (green) bounding boxes.

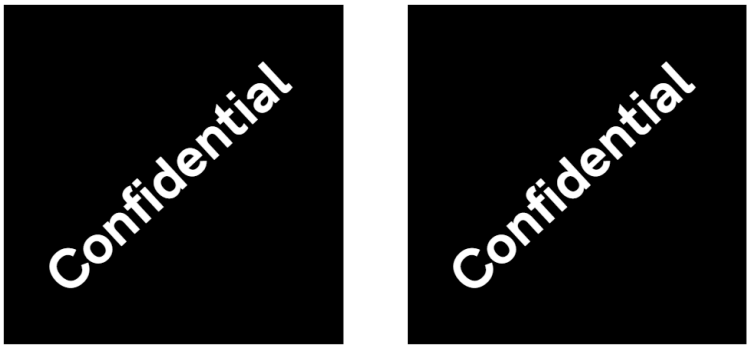
5.4.1 Cropped Dataset

For integration with the classification models, regions of interest (ROI) corresponding to detected bounding boxes were extracted from the images.

For defect images, crops were generated automatically using YOLOv8 predictions, forming the input to the classification stage of the pipeline. Images labeled as "No Defect" contained no bounding boxes and therefore were not cropped at this stage. Figures 5.5(a) and 5.5(b) illustrate an example of an image before and after cropping.

To maintain uniformity in the classification stage and avoid trivial correlations (e.g., full images implying "no defect"), the *None* class was also cropped with a random cropping strategy - A custom `RandomCropTransform`.

The cropped dataset was aligned with the original dataframe so that each cropped image inherited its label (defect type) from the parent sample. Oversampled variants of



(a) Original image (b) Cropped region

Figure 5.5: Example of an image before and after cropping.

defect images were also carried forward. This ensured that the cropped dataset and its metadata remained consistent with the full dataset.

5.5 Defect Classification Models

For the classification task, several deep learning architectures were explored, ranging from lightweight convolutional networks to modern transformer-based models. Given the limited and imbalanced dataset, transfer learning with pre-trained models on ImageNet-21k was employed to leverage rich feature representations while mitigating the risk of severe overfitting.

This section outlines the strategies adopted to address class imbalance, prevent data leakage, and ensure fair comparison across models, as well as the optimization and fine-tuning procedures applied to each architecture.

5.5.1 Dataset Balancing

To address the severe class imbalance described in Section 4.2, the dominant *None* class was undersampled. The target ratio of defects to non-defects was set to approximately 40%. Table 5.1 summarizes the class distribution before and after undersampling in the original dataset.

Table 5.1: Value counts per defect after undersampling (Original Dataset).

Label	**Before undersampling**	**After undersampling**
None	1024	117
Weld Overlay	27	27
Bead Height	24	24
Lack of Fusion	24	24
Porosity	3	3
Total	1102	195

In the cropped dataset, defect images were approximately five times more numerous due to oversampling during detection model training.

The two variants were compared:

- **Option 1 – With Oversampling:** Higher weighted F1 and accuracy, but poorer performance on minority defects, as reflected by macro F1.
- **Option 2 – Without Oversampling:** Lower overall accuracy, but better per-class performance for minority defects, which is more meaningful for defect detection.

Therefore, to also ensure comparability, undersampling was applied, so both cropped and original datasets contained the same number of images per class. Table 5.2 shows the resulting distributions.

Table 5.2: Value counts per defect after undersampling (Cropped Dataset).

Label	Before undersampling	Option 1	Option 2
None	1024	502	118
Weld Overlay	123	123	27
Bead Height	117	117	25
Lack of Fusion	84	84	24
Porosity	11	11	3
Total	1359	837	197

In practice, undersampling raised the proportion of defect images from 7.1% to roughly 40%. However, it also reduced dataset size, increasing the risk of underfitting.

No data leakage strategy

To ensure reliable evaluation, a strict no data leakage approach was enforced by assigning images from sequential frames of the same seam to the same fold. This avoided visually similar samples from appearing in both training and validation sets, which would otherwise inflate results.

For the minority *Porosity* class, which contained only three samples (sequential images of the same weld), this meant no validation examples were available. Consequently, metrics for *Porosity* are absent in leakage-free runs. To ensure fairness, results include comparisons with other strategies that allowed evaluation of this class (e.g., possible data leakage).

5.5.2 Model Architectures

The following model architectures were employed for the defect classification task:

- **ResNet18** [22]: Served as a lightweight CNN baseline, offering fast training and low overfitting risk. Its shallow design required fewer data, making it well-suited to the limited dataset.

- **ResNet50** [22]: A deeper CNN baseline with higher representational capacity, but also greater susceptibility to overfitting.
- **ConvNeXt** [26]: A modernized CNN architecture inspired by Transformers, emphasizing efficient convolutional design. Both Tiny and Small variants were tested, with Small outperforming the Tiny variant, which struggled on minority classes.
- **Swin Transformer (Swim)** [25]: Employs hierarchical vision transformers with shifted windows, balancing computational efficiency and global context. Tiny and Base variants were tested; Tiny was retained due to the overfitting of the Base model.
- **Vision Transformer (ViT)** [24]: Applies self-attention to image patches to capture global dependencies. Tiny and Small variants were tested; Tiny performed slightly better under no-leakage training. However, ViT underperformed relative to other architectures, likely due to dataset size limitations.

5.5.3 Experimental Framework

All classification models were trained under a consistent experimental framework. Table 5.3 summarizes the baseline training parameters, which ensured fair comparison across architectures.

Table 5.3: Baseline training parameters for classification models.

Parameter	Value
Cross-validation	3-fold ($k = 3$)
Max epochs	20
Early stopping	Patience = 7
Batch size	32
Augmentation	Class-specific (baseline)
Class weights	Enabled
Undersampling rate	≈ 0.4 (main tests)

In addition, four training strategies were compared (applied to both datasets):

1. MixUp,
2. Conditional MixUp (minority classes only),
3. Conditional MixUp + No Data Leakage,
4. No Data Leakage.

This resulted in 8 experimental configurations per model (4 strategies $\times$ 2 datasets). Configurations 3 and 4 explicitly prevented data leakage by ensuring multiple variants of the same image did not appear across training and validation folds, while configurations 1 and 2 were retained for comparison with earlier experiments.

Class-specific weighting was applied in all settings to mitigate imbalance. Undersampling was initially tested at rates between 0.2 and 0.3, but the main experiments finalized at 0.4, balancing class proportions with the dataset size.

Beyond these fixed settings, model-specific hyperparameters such as the choice of loss function, optimizer, scheduler, and learning rates were adapted as described below.

5.5.4 Optimization and Fine-Tuning

To ensure fair and effective training across architectures, a consistent optimization framework was adopted and then adapted to the specific requirements of each model. This involved careful selection of loss functions, optimizers, and learning rate schedulers, as well as strategies for gradual fine-tuning of deeper networks. The goal was to stabilize training while allowing each architecture to leverage its representational capacity without severe overfitting. The main components of this framework are outlined below.

Loss functions: The primary loss was *CrossEntropyLoss*, often with label smoothing ($\alpha = 0.1$) to reduce overconfidence. *Focal Loss* was tested for class imbalance mitigation but degraded minority class performance and was not retained for final experiments.

Optimizers: Adam and AdamW were the main optimizers. Adam was used for ResNets, while AdamW was preferred for ConvNeXt and Transformer-based models due to its stability in fine-tuning.

Schedulers: Learning rate schedules varied by model: for ResNets, either `ReduceLROnPlateau` or `StepLR` was used, while for ConvNeXt, Swin Transformer, and Vision Transformer, `CosineAnnealingWarmRestarts` ($T_0 = 5$, $T_{\text{mult}} = 2$) was applied.

Learning rates: Learning rates typically ranged between 1×10^{-4} and 5×10^{-5}. For Transformer-based models, differentiated rates were applied: higher for classification heads (1×10^{-4}) and lower for backbone stages (5×10^{-5} or less).

Phased fine-tuning: For deeper architectures (ConvNeXt, Swin Transformer, and Vision Transformer), a three-phase fine-tuning strategy was employed, progressively unfreezing layers with different learning rates per parameter group:

- **Phase 1:** Train classification head only.
- **Phase 2:** Unfreeze final backbone stage with lower learning rate.
- **Phase 3:** Fully unfreeze backbone for final fine-tuning (mainly Cropped Dataset and ensemble experiments).

This approach provided controlled flexibility while preserving comparability across models.

Final hyperparameters

Table 5.4 summarizes the best-performing hyperparameters for each model.

All models shared the baseline setup (Table 5.3), with model-specific choices in loss functions, optimizers, schedulers, and fine-tuning phases. ResNet18 was trained end-to-end with a dropout of 0.6 applied to the fully connected layer to improve generalization, while ResNet50 benefited from label smoothing and cosine annealing schedulers to improve stability on the limited dataset. Among the modern architectures, ConvNeXt-Small employed a two-phase fine-tuning strategy, Swin Transformer-Tiny adopted progressive unfreezing that enhanced minority class detection, and ViT-Tiny relied on three-phase fine-tuning but showed more limited performance due to dataset size.

Overall, the models were trained under a consistent baseline framework, with tailored optimization and fine-tuning strategies designed to ensure fairness and enable robust comparison across architectures.

Table 5.4: Best hyperparameters for each model.

Model	**Best Hyperparameters**
ResNet18	LR: 0.0004 Loss: CrossEntropyLoss (label smoothing = 0.1) Optimizer: Adam Scheduler: ReduceLROnPlateau, factor = 0.5, patience = 2 FC layer: Dropout(0.6) + Linear
ResNet50	LR: 0.0001 Loss: CrossEntropyLoss (label smoothing = 0.1) Optimizer: Adam Scheduler: CosineAnnealingWarmRestarts, $T_0 = 5$, $T_{mult} = 2$
ConvNeXt - Small	Phase: 2 LRs: head = 1e-4, last stage = 5e-5 Loss: CrossEntropyLoss (label smoothing = 0.1) Optimizer: AdamW Scheduler: CosineAnnealingWarmRestarts, $T_0 = 5$, $T_{mult} = 2$
Swin Transformer - Tiny	Phase: 3 Loss: CrossEntropyLoss (label smoothing = 0.1) Optimizer: AdamW Scheduler: CosineAnnealingWarmRestarts, $T_0 = 5$, $T_{mult} = 2$
Vision Transformer - Tiny	Phase: 3 LRs: head = 1e-4, last stage = 5e-5 Loss: CrossEntropyLoss (label smoothing = 0.1) Optimizer: AdamW Scheduler: CosineAnnealingWarmRestarts, $T_0 = 5$, $T_{mult} = 2$

6

Results and Discussion

6.1 Introduction

This chapter presents the experimental results obtained from the proposed defect inspection framework and discusses their implications. The results are structured to follow the pipeline stages defined in Chapter 5.1. First, the performance of the detection stage is analyzed, followed by a detailed evaluation of the classification stage across different datasets, including a comparison between the original and cropped datasets to highlight the impact of focusing on defect regions.

Finally, the outcomes of the complete inspection pipelines are presented, including a comparative analysis of the Global, Classification-only, and Hybrid configurations, considering accuracy, robustness, and computational cost.

This structured evaluation allows for both model-specific and system-level comparisons, ensuring that performance gains can be attributed to specific design choices. Methodological factors, such as the effects of data augmentation and data leakage, are discussed in the Limitations and Observations section (Section 6.5).

6.2 Defect Detection Results

The detection stage was evaluated using YOLOv8n models trained under different data balancing and augmentation strategies. Both models employed geometric and photometric augmentations (rotation, translation, scaling, shear, flipping, HSV transformations, mosaic, and MixUp), with one model additionally leveraging oversampling and copy-paste augmentation.

Multi-Class Detection

The baseline YOLOv8n model was trained on the dataset without oversampling, which contained only 13 defect instances in validation across all four defect classes. Consequently, the model exhibited poor generalization, with some defect types remaining undetected. Table 6.1 summarizes the validation results.

Table 6.1: Detection results without oversampling (multi-class YOLOv8n).

Class	Precision	Recall	mAP@0.5	mAP@0.5:0.95
Bead Height	0.000	0.000	0.000	0.000
Lack of Fusion	1.000	0.654	0.669	0.502
Porosity	0.904	0.500	0.504	0.287
Weld Overlay	0.884	1.000	0.995	0.895
All classes	0.697	0.538	0.542	0.421

Detection performance was highly uneven: *Weld Overlay* achieved near-perfect detection, whereas *Bead Height* was never detected. This outcome reflects the scarcity of examples for certain classes.

Binary Detection

To mitigate imbalance, defect samples were oversampled fivefold and augmented using copy-paste. The task was also simplified to binary detection (*Defect* vs. *No Defect*). Table 6.2 presents the results of this model.

Table 6.2: Detection results with oversampling and copy-paste (binary YOLOv8n).

Class	Precision	Recall	mAP@0.5	mAP@0.5:0.95
Defect	0.980	1.000	0.995	0.670

Binary detection achieved near-perfect precision and recall, with an mAP@0.5 of 0.995. Performance decreased at stricter IoU thresholds (mAP@0.5:0.95 = 0.67), but remained a substantial improvement over the multi-class configuration.

Discussion

The experiments demonstrate the critical role of data balancing in defect detection. Without oversampling, minority defect types were undetected, whereas oversampling combined with binary detection provided robust and consistent performance across all defect types.

These results suggest that, for datasets with very few defect examples, multi-class detection is often infeasible. Instead, binary detection serves as an effective first-stage filter for downstream classification.

Test Set Evaluation: On the held-out test set, the binary YOLOv8n model achieved perfect metrics: Accuracy = 1.0, Precision = 1.0, Recall = 1.0, F1 = 1.0. All defect and non-defect instances were correctly identified, confirming the reliability of the detection stage. Figure 6.1 shows the corresponding confusion matrix, highlighting the flawless detection across both categories.

It is important to note, however, that these perfect results are likely influenced by the small size of the test dataset and do not necessarily reflect the absolute reliability of the model in broader deployment scenarios.

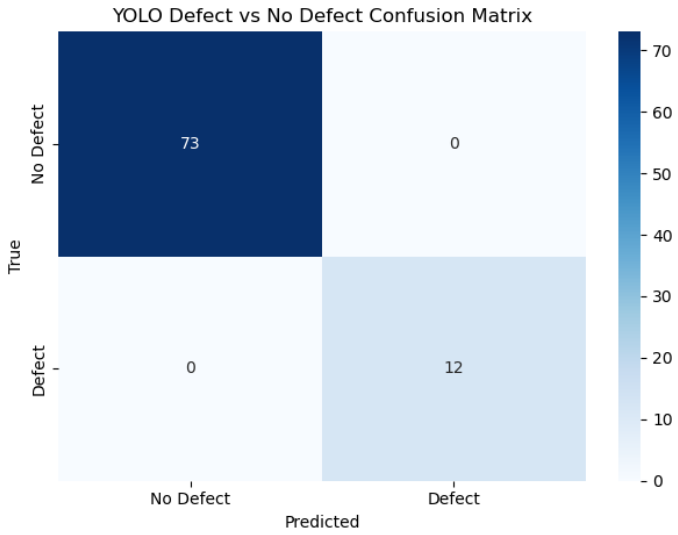


Figure 6.1: Confusion matrix for the binary YOLOv8n model on the test dataset.

6.3 Defect Classification Results

After the detection stage, classification was evaluated on both the *original* and *cropped* datasets. Models (ResNet18, ResNet50, Swin - Tiny, ViT - Tiny, ConvNeXt - Small) were trained under four strategies varying augmentation and leakage prevention:

1. Class-specific augmentation with MixUp,

2. Conditional MixUp applied to minority classes,

3. Conditional MixUp with leakage prevention,

4. Class-specific augmentation without MixUp and with leakage prevention.

Performance was primarily evaluated using macro-F1 score, as well as per-class F1 and support. Detailed classification results for all models and for each strategy on the original and cropped datasets are provided in Appendix B.

A compact version of this tables are presented in this section just for clarity and simplicity.

Notably, in the no-leakage setting (Strategies 3 and 4), the class *Porosity* had no support due to the split procedure (see Subsection 5.5.1), which inevitably reduced the macro-F1 score.

6.3.1 Original Dataset

Table 6.3 summarizes macro-F1 scores. Strategy 1 (MixUp) consistently outperformed others, while leakage-free strategies (3 and 4) showed reduced scores, especially for minority classes.

Table 6.3: Macro F1-scores on the **original dataset**.

Model	Strategy 1	Strategy 2	Strategy 3	Strategy 4
ResNet18	0.694	0.588	0.330	0.413
ResNet50	0.737	0.388	0.182	0.179
Swin-Tiny	0.746	0.410	0.175	0.316
ViT-Tiny	0.613	0.451	0.201	0.441
ConvNeXt-Small	0.651	0.310	0.247	0.230

Performance was modest (macro-F1 rarely above 0.75), likely due to background noise and dataset imbalance. Leakage artificially inflated apparent performance, particularly when MixUp was applied across folds.

6.3.2 Cropped Dataset

The cropped dataset substantially improved performance (Table 6.4), with macro-F1 exceeding 0.95 for ConvNeXt and Swin Transformer under MixUp-based augmentations. Leakage-free strategies (3 and 4) remained competitive, demonstrating that focusing on defect regions enhances robustness.

Table 6.4: Macro F1-scores on the **cropped dataset**.

Model	Strategy 1	Strategy 2	Strategy 3	Strategy 4
ResNet18	0.891	0.792	0.465	0.528
ResNet50	0.974	0.848	0.164	0.503
Swin-Tiny	0.974	0.980	0.536	0.614
ViT-Tiny	0.343	0.369	0.209	0.458
ConvNeXt-Small	0.983	0.959	0.523	0.646

Analysis of the results indicates that indicate that MixUp was generally effective in improving class generalization, particularly for ConvNeXt and Swin Transformer models. However, this technique did not yield improvements when strict no-data-leakage strategies were enforced. Architectures with strong inductive biases, such as ResNets, benefited more from cropping than from data augmentation alone. In contrast, the Vision Transformer consistently underperformed, likely due to the limited dataset size and the reduced contextual information present in the cropped images.

Discussion and Comparison

Comparing the datasets, the original dataset yielded modest performance, with minority classes poorly predicted, whereas the cropped dataset substantially improved overall accuracy and robustness. Leakage-free strategies (Strategies 3 and 4) provide the fairest evaluation, even though they result in lower macro-F1 scores, while MixUp tended to inflate performance when data leakage might have been present.

For downstream pipeline evaluation, Strategy 4 (no MixUp, leakage-free) was adopted as it provides the most reliable and unbiased basis for model assessment. On the original dataset, the combination of ViT-Tiny and ResNet18 achieved the best overall performance, particularly improving predictions for minority classes. On the cropped dataset, ConvNeXt emerged as the top-performing model, offering a balance between high overall accuracy and robust handling of challenging defect classes.

These experiments highlight two important insights. First, cropping defect regions significantly improves classification by reducing background noise. Second, while MixUp can be effective in mitigating class imbalance, it may hinder performance under a strict no-data-leakage regime.

6.4 Pipeline Performance

Following the defect classification analysis, the performance of the complete inspection pipelines was evaluated.

As described in Section 5.1, three pipeline configurations were considered: the **Global Pipeline**, which sequentially performs detection and classification using a unified model; the **Classification Pipeline**, which focuses on classifying original images using Strategy 4 models; and the **Hybrid Pipeline**, which combines multiple models to optimize detection and classification, particularly for minority defect classes.

This section presents the results for each pipeline configuration, highlights their relative strengths, and discusses trade-offs in terms of accuracy, robustness, and class-specific performance.

The goal of this analysis is to determine the most effective pipeline—and the most suitable model within each pipeline—for practical deployment, ensuring reliable detection and classification across all defect types while mitigating the effects of class imbalance and dataset biases.

6.4.1 Global Pipeline Results

The Global Pipeline integrates the defect detection stage with classification models selected from the cropped dataset (Strategy 4). Based on the latest evaluation, ConvNeXt emerged as the best-performing model, both individually and in combination with ResNet18, achieving the highest F1 score and overall accuracy.

Three configurations were compared within the Global Pipeline: YOLOv8n + ConvNeXt-Small; YOLOv8n + ConvNeXt-Small + ResNet18; and YOLOv8n + Swin-Tiny (included for comparison).

Table 6.5 summarizes the best performance of each configuration, including the confidence threshold that maximized F1 score, overall accuracy, and macro-F1.

Table 6.5: Global pipeline performance across configurations. Best thresholds, overall Accuracy, and F1 are reported.

Configuration	Best Threshold	Accuracy	F1 (macro)
YOLOv8n + ConvNeXt-Small	0.20	0.9647	0.7450
YOLOv8n + ConvNeXt-Small + ResNet18	0.20	0.9647	0.7450
YOLOv8n + Swin-Tiny	0.80	0.9294	0.4901

The YOLOv8n + ConvNeXt-Small pipeline achieved the highest F1 score (0.745), demonstrating strong performance across majority classes and several minority defect classes. Swin-Tiny, which had been previously competitive, now exhibits a lower F1 score (0.4901) and is clearly outperformed by ConvNeXt.

The results obtained with YOLOv8n + ConvNeXt-Small and YOLOv8n + ConvNeXt-Small + ResNet18 were identical for both the Global and Hybrid pipelines. To simplify presentation, only the YOLOv8n + ConvNeXt-Small configuration will be discussed, although both models perform equivalently. Consequently, the model adopted for the Global Pipeline is YOLOv8n + ConvNeXt-Small.

Threshold Sweep

A confidence threshold sweep was conducted to determine the optimal operating point for the YOLOv8n + ConvNeXt-Small pipeline.

Figure 6.2 presents the accuracy and F1 score at each threshold. The best performance was achieved at a confidence threshold of 0.20, with an overall Accuracy of 0.9647 and F1 score of 0.7450.

Table 6.6: Threshold sweep results for the YOLOv8n + ConvNeXt-Small pipeline.

Confidence Threshold	Accuracy	F1 (macro)
0.10	0.9412	0.6790
0.20	0.9647	**0.7450**
0.30	0.9647	0.7450
0.40	0.9647	0.7450
0.50	0.9647	0.7450
0.60	0.9647	0.7450
0.70	0.9647	0.7450
0.80	0.9294	0.4901
0.90	0.9294	0.4901

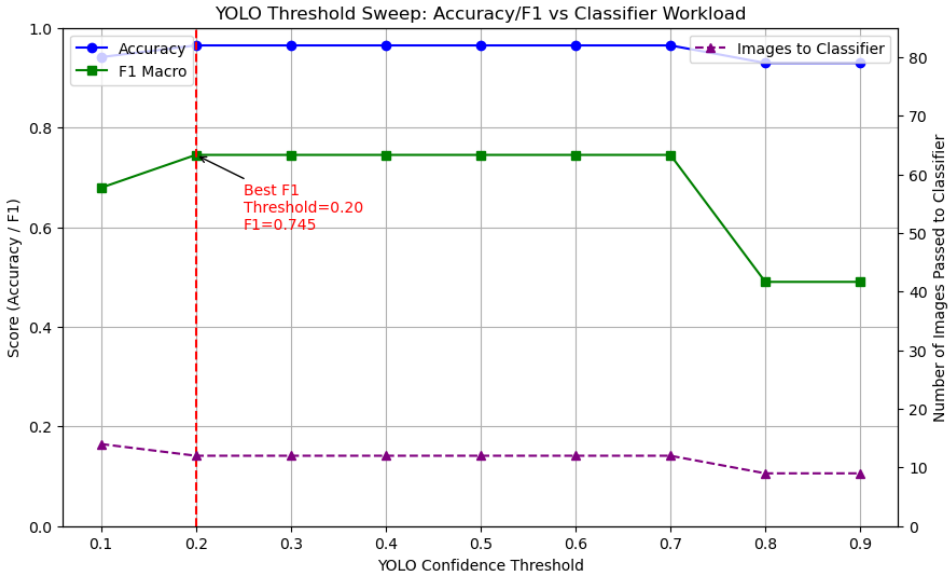


Figure 6.2: Threshold Sweep Results.

Performance stability across thresholds from 0.20 to 0.70 indicates that the pipeline is generally resilient to threshold variation, with notable degradation occurring only at very strict thresholds (≥ 0.80). In particular, the lower threshold of 0.20 allows a greater number of images to be classified correctly, improving detection and classification of minority defects, such as *Lack of Fusion*, while maintaining high accuracy for majority classes.

Per-class Performance at Best Threshold

Table 6.7 presents per-class precision, recall, F1 score, and support for the pipeline at the optimal threshold of 0.20.

The results indicate that the pipeline achieves high precision and recall for the majority class (*None*) as well as for some minority classes (*Lack of Fusion* and *Weld Overlay*). The *Bead Height* class remains challenging due to very limited support in the test set.

Table 6.7: Per-class performance for YOLOv8n + ConvNeXt-Small at confidence threshold 0.20.

Class	Precision	Recall	F1	Support
Bead Height	0.000	0.000	0.000	3
Lack of Fusion	1.000	1.000	1.000	3
None	0.961	1.000	0.980	73
Porosity	0.000	0.000	0.000	0
Weld Overlay	1.000	1.000	1.000	6

The confusion matrix at the optimal threshold, shown in Figure 6.3, provides a detailed visualization of correct and incorrect predictions. These results confirm that YOLOv8n + ConvNeXt-Small is the preferred configuration for the Global Pipeline, achieving the

highest overall F1 score and strong performance across most defect classes, while also highlighting the remaining challenges associated with extremely rare defects.

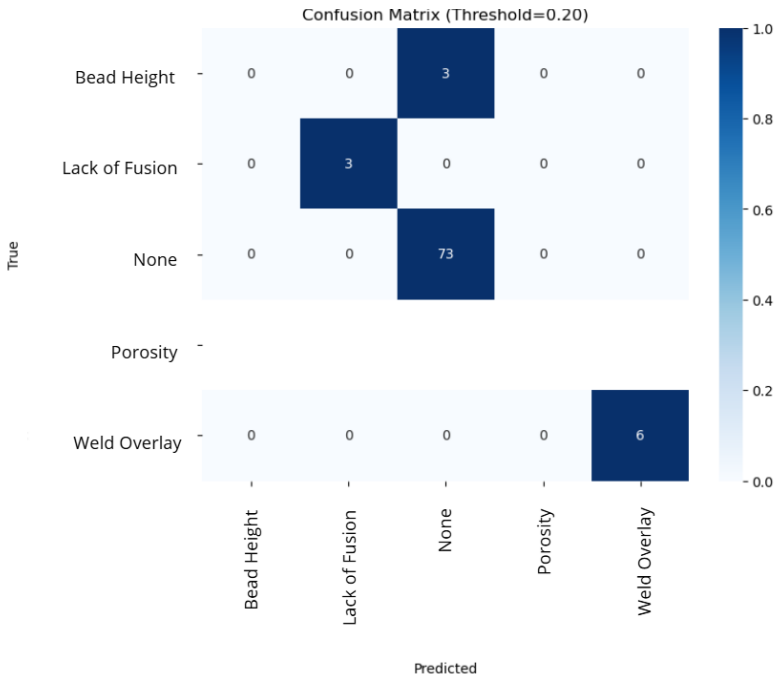


Figure 6.3: Confusion Matrix of YOLOv8n + ConvNeXt-Small Pipeline at Best Threshold.

It is also worth noting that excluding the *Bead Height* class from evaluation increases the overall F1 score to 0.82 and the overall accuracy to 0.97 at a threshold of 0.20, underscoring that the low performance for this class is primarily due to its limited test representation rather than inherent model limitations.

6.4.2 Classification Pipeline

The classification pipeline was evaluated using models trained on the *original dataset*, focusing on ResNet18, ViT-Tiny, ConvNeXt-Small, and Swin-Tiny. Each model was tested individually, followed by ensemble configurations to assess whether combining predictions improved performance. All evaluations were performed on the same test set used for the Global Pipeline.

Table 6.8 summarizes the overall Accuracy and macro-F1 for each model and ensemble. The ConvNeXt-Small + Swin-Tiny ensemble achieved the best overall performance, reaching an Accuracy of 0.8235 and a macro-F1 score of 0.5560. This suggests that combining complementary convolutional and transformer-inspired architectures yields more robust performance than individual models or simpler ensembles, such as ResNet18 + ViT-Tiny.

Table 6.8: Overall performance of classification models on the original dataset.

Model	Accuracy	Macro F1
ResNet18	0.4471	0.3112
ViT-Tiny	0.4941	0.3080
ConvNeXt-Small	0.7647	0.3749
Swin-Tiny	0.7529	0.4617
ResNet18 + ViT-Tiny Ensemble	0.5294	0.3640
ConvNeXt-Small + Swin-Tiny Ensemble	**0.8235**	**0.5560**
Full Ensemble (ResNet18+ViT-Tiny+ConvNeXt-Small+Swin-Tiny)	0.5882	0.4085

Table 6.9 provides per-class metrics for the ConvNeXt-Small + Swin-Tiny ensemble. While classes such as *None* and *Weld Overlay* achieved high scores, minority classes like *Lack of Fusion* and *Porosity* remained challenging due to very limited support in the test set.

Table 6.9: Per-class metrics for the ConvNeXt-Small + Swin-Tiny ensemble on the original dataset.

Class	Precision	Recall	F1	Support
Bead Height	0.200	1.000	0.333	3
Lack of Fusion	0.000	0.000	0.000	3
None	0.953	0.836	0.891	73
Porosity	0.000	0.000	0.000	0
Weld Overlay	1.000	1.000	1.000	6

These results demonstrate that modern deep learning architectures can achieve strong overall performance; however, they continue to struggle with extremely imbalanced classes. The ConvNeXt-Small + Swin-Tiny ensemble outperformed all other configurations, confirming that ensembles of complementary architectures provide more reliable predictions. Nevertheless, some minority defects remained undetected, highlighting the limitations imposed by insufficient training samples.

Overall, the classification pipeline underperformed relative to the Global Pipeline, confirming that leveraging localized cropped regions provides a stronger foundation for defect classification.

6.4.3 Hybrid Pipeline

The Hybrid Pipeline integrates the Global Detection Pipeline (YOLOv8n + ConvNeXt-Small) with a fallback mechanism to the Classification Pipeline (ConvNeXt-S + Swin-Tiny ensemble).

In the Global Pipeline, images without detected defects are automatically classified as *None*. In contrast, the Hybrid Pipeline passes these images to the classification ensemble, enabling the system to potentially recover minority defect detections that may have been

missed by YOLOv8n. The Hybrid Pipeline was evaluated across YOLOv8n confidence thresholds from 0.10 to 0.90.

Table 6.10 reports overall Accuracy and macro-F1 for each threshold, with the best performance observed at a threshold of 0.20, achieving an Accuracy of 0.8235 and a macro-F1 of 0.7226. Per-class performance at this optimal threshold is detailed in Table 6.11.

Table 6.10: Threshold sweep results for the Hybrid Pipeline (YOLOv8n + ConvNeXt-Small with classification fallback).

Threshold	Accuracy	Macro F1
0.10	0.8235	0.6601
0.20	0.8235	**0.7226**
0.30	0.8235	0.7226
0.40	0.8235	0.7226
0.50	0.8235	0.7226
0.60	0.8235	0.7226
0.70	0.8235	0.7226
0.80	0.7882	0.4679
0.90	0.7882	0.4679

Table 6.11: Per-class metrics for the Hybrid Pipeline at confidence threshold 0.20.

Class	Precision	Recall	F1	Support
Bead Height	0.000	0.000	0.000	3
Lack of Fusion	1.000	1.000	1.000	3
None	0.953	0.836	0.891	73
Porosity	0.000	0.000	0.000	0
Weld Overlay	1.000	1.000	1.000	6

The Hybrid Pipeline substantially improves detection of minority classes, most notably *Lack of Fusion*, which reached a perfect F1 score. *Weld Overlay* was also consistently detected, while the *None* class retained high performance. By combining YOLO-based localization with a fallback to the ConvNeXt-Small + Swin-Tint ensemble, the Hybrid Pipeline successfully recovers minority-class detections missed by the Global Pipeline, without compromising the performance of majority classes.

However, this recovery comes at a cost. The *Bead Height* class, partially detected by the Global Pipeline, is not captured by the fallback mechanism, and *Porosity* remains undetected in both pipelines due to the absence of support in the test set.

Overall, the Hybrid Pipeline does not strictly outperform the Global Pipeline in terms of accuracy or macro-F1. Rather, it redistributes minority-class coverage, improving recall for some classes while reducing it for others. The confusion matrix in Figure 6.4 provides a clear illustration of this trade-off: fallback strategies can enhance detection of specific minority defects but may inadvertently reduce coverage of classes already captured by YOLOv8n.

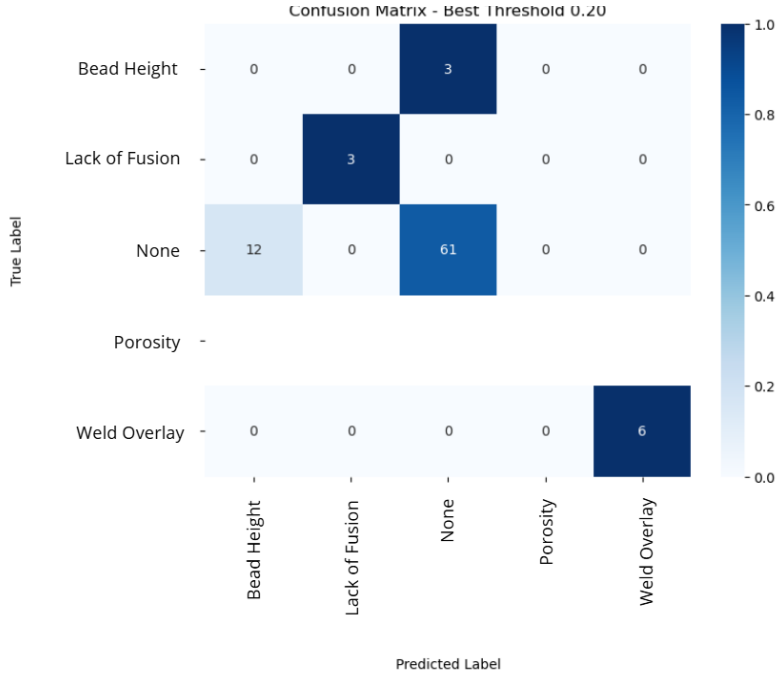


Figure 6.4: Confusion matrix for the Hybrid Pipeline at confidence threshold 0.20.

6.4.4 Comparative Analysis

Table 6.12 summarizes the performance of the three evaluated pipelines. The Global Pipeline, combining YOLOv8n + ConvNeXt-Small, achieved the highest macro-F1 and the best overall accuracy.

Table 6.12: Summary of evaluated pipelines.

Pipeline	Accuracy	Macro F1	Notes
Global (YOLOv8n+ConvNeXt)	**0.9647**	**0.7450**	Best overall
Classification-only (ConvNeXt+Swin)	0.8235	0.5560	Moderate performance
Hybrid (Global + fallback)	0.8235	0.7226	Better minority recall

This comparative analysis highlights several key observations:

- **Detection stage is crucial:** YOLOv8n cropping reduces background noise, enabling superior accuracy and macro-F1.
- **Model selection matters:** ConvNeXt-Small provides strong single-model performance, while Swin-Tiny complements it in ensembles to enhance minority-class detection.
- **Hybrid pipeline:** offers some improvement in minority-class recall through fallback classification, but its overall macro-F1 remains below that of the Global Pipeline.

The fallback classifier successfully recovers *Lack of Fusion* but loses detection of *Bead Height*, illustrating that gains in coverage for some minority classes may come at the expense of others.

Overall, the Global Pipeline achieves the best performance in terms of both accuracy and macro-F1, while the Hybrid Pipeline can be useful for capturing additional minority-class instances at a small cost to overall metrics.

6.5 Limitations and Observations

The main challenges in this work stemmed from extreme class imbalance, limited dataset size, and the sensitivity of models to dataset preparation and augmentation strategies. While the Global Pipeline (YOLOv8n + ConvNeXt-Small) achieved the highest overall performance, all pipelines were fundamentally constrained by the scarcity of defect samples.

Data augmentation played a critical role in enabling model training on the limited dataset. Techniques such as MixUp, conditional MixUp, class-specific augmentation, and cropping enriched the data and improved generalization. Nonetheless, important limitations persisted. Extreme class imbalance could not be fully mitigated, and MixUp's effectiveness proved context-dependent—beneficial under potential data leakage but detrimental in leakage-free setups. Cropping improved classifier focus by reducing background noise, but it also reduced dataset variability, negatively affecting context-dependent models such as ViT and Swin.

Data leakage was another significant factor. Initial uncontrolled splits inflated performance due to near-duplicate images appearing in both training and validation sets, producing biased majority-class predictions and artificially high F1 scores for minority classes. Enforcing strict leakage-free splits revealed the true difficulty of predicting rare defect classes, with some classes absent from validation or achieving zero F1 scores.

Detection and classification models faced complementary challenges. YOLO-based detection benefited from object-level augmentations such as Mosaic and Copy-Paste and even reported perfect metrics; however, this is likely an artifact of very few test images rather than true performance.

Classification models improved when images were cropped to reduce background noise, yet small dataset size limited the effectiveness of transformer-based architectures. Ensemble methods provided only incremental gains, insufficient to overcome the fundamental limitations imposed by data scarcity.

Comparing pipeline configurations highlighted clear trends. The Global Pipeline consistently outperformed alternatives by effectively combining detection and classification, whereas hybrid pipelines and ensemble strategies offered only minor improvements,

mainly in minority-class recall. Threshold selection was critical to balance false positives and missed defects.

Several broader limitations must be acknowledged:

- Extremely scarce defect classes, such as *Porosity*, restricted achievable model performance.
- Variations in lighting, weld orientation, and sensor noise may reduce generalization in real-world conditions.
- Reliance on synthetic augmentation could bias models toward artifacts rather than true defect characteristics.
- The Vision Transformer underperformed due to the limited dataset size, suggesting the need for additional data for this model to be able to perform adequately.
- Macro-F1 metrics, while useful for assessing model performance, are highly sensitive to very rare classes, where a single misclassification can heavily impact scores.

Based on these observations, several practical recommendations emerge. Leakage-free dataset splits are essential for realistic evaluation. Combining detection and classification stages, as in the Global Pipeline, yields more robust performance. Cropping should be applied carefully to reduce background noise while preserving dataset variability. Detection thresholds should be tuned empirically, and augmentation for minority classes must be applied judiciously to avoid artificially inflating performance.

7

Conclusions and Future Work

7.1 Conclusions

This work investigated the problem of surface defect detection and classification in welded components using computer vision techniques. The study involved designing an image support and acquisition system, curating a dedicated dataset, preprocessing the images, implementing multiple deep learning models, and systematically evaluating augmentation strategies and data handling procedures.

The experiments highlighted the critical role of dataset preparation in model performance. Cropping images around the weld seam significantly reduced background noise, and further cropping around the defect itself improved the detection and classification of subtle defects. Rigorous prevention of data leakage was essential for obtaining realistic performance estimates, emphasizing the importance of careful dataset splitting.

Among the models evaluated, YOLO-based detection models effectively localized all defects and non-defect instances, benefiting strongly from object-level augmentations. Notably, if the task were framed solely as detection, the YOLOv8n model would have achieved perfect results, correctly predicting every defect instance. However, when extended to classification, performance became constrained by dataset imbalance and the difficulty of distinguishing between visually similar defect categories.

Convolutional architectures—particularly ConvNeXt—achieved the strongest performance on cropped regions, while Vision Transformers struggled due to the limited dataset size.

The best overall results were obtained with the Global Pipeline (YOLOv8n + ConvNeXt-Small), achieving the highest accuracy (0.9647) and macro-F1 score (0.7450). The Hybrid Pipeline, combining the Global Pipeline with the Classification-only ConvNeXt-Small + Swin-Tiny ensemble as a fallback, provided more balanced performance for minority classes, particularly recovering *Lack of Fusion*, but at the cost of missing other classes, such as *Bead Height*.

The classification-only pipeline performed moderately well (Accuracy 0.8235, macro-F1 0.5560) but struggled with minority defects. These results underscore that combining

localized defect detection with appropriate classification models is crucial for robust performance.

Despite these advances, performance on rare defect classes remains limited, highlighting the inherent difficulty of reliably identifying minority welding defects with small, imbalanced datasets. Data augmentation techniques, such as MixUp, can help mitigate this issue but cannot fully compensate for severe scarcity.

In general, this study demonstrates that high-quality image acquisition systems, rigorous dataset design, effective preprocessing, appropriate evaluation protocols, and careful model selection are critical components for the success of an automated visual inspection system.

7.2 Suggestions for future works

The work developed in this dissertation also presents opportunities for improvement. Thus, some suggestions for future developments are presented:

1. Expanding the dataset by collecting additional images for underrepresented defect classes would enhance model generalization and robustness.

2. Integrating systems for automated component positioning during image acquisition could reduce variability and improve inspection efficiency.

3. Advanced augmentation techniques, including generative models, domain adaptation, and synthetic data generation, such as GANs, could help mitigate class imbalance and enrich scarce defect classes.

4. Pipeline and model optimizations that include investigating additional architectures, ensemble strategies, and adaptive thresholding, as well as exploring transformer-based models fine-tuned for small datasets to enhance detection and classification of minority classes.

5. Real-time deployment considerations are also crucial; although current models achieve high accuracy, they are computationally intensive, and reducing inference latency will be essential for practical AVI deployment.

6. Field testing and validation in operational environments will provide insights into robustness, latency, and usability, while identifying areas for further improvement.

Bibliography

[1] J. M. Lourenço. *The NOVAthesis LaTeX Template User's Manual*. NOVA University Lisbon. 2021. URL: https://github.com/joaomlourenco/novathesis/raw/main/template.pdf (cit. on p. i).

[2] S. S. B. M. Hussain et al. *Deep Learning-Enhanced Augmented Reality for Real-Time Welding Training and Defect Detection*. https://ssrn.com/abstract=5088028. Available at SSRN: http://dx.doi.org/10.2139/ssrn.5088028. 2024 (cit. on p. 3).

[3] J. Beyerer, F. Puente León, and C. Frese. *Machine Vision: Automated Visual Inspection: Theory, Practice and Applications*. 1st ed. Springer-Verlag Berlin Heidelberg, 2016, p. 798. ISBN: 978-3-662-47793-9. DOI: 10.1007/978-3-662-47794-6 (cit. on pp. 3, 7–9).

[4] H. Golnabi and A. Asadpour. "Design and Application of Industrial Machine Vision Systems". In: *Robotics and Computer-Integrated Manufacturing* 23.6 (2007), pp. 630–637. DOI: 10.1016/j.rcim.2007.02.005 (cit. on p. 3).

[5] T. Ozseven. "Surface Defect Detection and Quantification with Image Processing Methods". In: *Theoretical Investigations and Applied Studies in Engineering*. Ed. by T. Ozseven. Ekin Publishing House, 2019, pp. 63–98 (cit. on p. 3).

[6] X. Zheng, S. Zheng, Y. Kong, et al. "Recent Advances in Surface Defect Inspection of Industrial Products Using Deep Learning Techniques". In: *International Journal of Advanced Manufacturing Technology* 113 (2021), pp. 35–58. DOI: 10.1007/s00170-021-06592-8 (cit. on pp. 4, 8, 10, 18, 20).

[7] N. R. Mandal. "Ship Construction and Welding". In: *Springer Series on Naval Architecture, Marine Engineering, Shipbuilding and Shipping*. 2. Springer Nature Singapore Pte Ltd., 2017. DOI: 10.1007/978-981-10-2955-4_19 (cit. on p. 4).

[8] D. Say et al. *Artificial Intelligence Assistive Non-Destructive Testing of Welding Joints: A Review*. 2024-10. DOI: 10.20944/preprints202410.1518.v1 (cit. on pp. 4, 5, 17).

[9] A. Mohammed and M. Hussain. "Advances and Challenges in Deep Learning for Automated Welding Defect Detection: A Technical Survey". In: *IEEE Access* (2025). Received 14 May 2025, accepted 21 May 2025, published 27 May 2025, current version 4 June 2025. DOI: 10.1109/ACCESS.2025.3574083 (cit. on pp. 4, 16).

[10] S. K. Dwivedi, M. Vishwakarma, and P. Soni. "Advances and Researches on Non Destructive Testing: A Review". In: *Materials Today: Proceedings* 5.2, Part 1 (2018). 7th International Conference of Materials Processing and Characterization, March 17-19, 2017, pp. 3690–3698. ISSN: 2214-7853. DOI: https://doi.org/10.1016/j.matpr.2017.11.620. URL: https://www.sciencedirect.com/science/article/pii/S2214785317328936 (cit. on pp. 5, 7).

[11] A. S. Madhvacharyula, A. V. S. Pavan, S. Gorthi, et al. "In Situ Detection of Welding Defects: A Review". In: *Weld World* 66 (2022), pp. 611–628. DOI: 10.1007/s40194-021-01229-6 (cit. on pp. 5–7).

[12] N. Hütten et al. "Deep Learning for Automated Visual Inspection in Manufacturing and Maintenance: A Survey of Open-Access Papers". In: *Applied System Innovation* 7.1 (2024), p. 11. DOI: 10.3390/asi7010011 (cit. on pp. 7, 8, 12, 16, 17, 20).

[13] C. G. Drury. "Inspection of Sheet Materials — Model and Data". In: *Human Factors* 17.3 (1975), pp. 257–265. DOI: 10.1177/001872087501700305 (cit. on p. 7).

[14] M. Janóczki et al. "Automatic Optical Inspection of Soldering". In: *Materials Science - Advanced Topics*. Ed. by Y. Mastai. London: IntechOpen, 2013. Chap. 16. DOI: 10.5772/51699. URL: https://doi.org/10.5772/51699 (cit. on pp. 7, 9).

[15] B. Tang et al. "Review of Surface Defect Detection of Steel Products Based on Machine Vision". In: *IET Image Processing* (2022). Open access article under the Creative Commons Attribution-NonCommercial License. DOI: 10.1049/ipr2.12647 (cit. on pp. 8–10, 18).

[16] I. Gräßler et al. "Creating Synthetic Training Data for Machine Vision Quality Gates". In: *Bildverarbeitung in der Automation*. Ed. by V. Lohweg. Berlin, Heidelberg: Springer Berlin Heidelberg, 2023, pp. 95–106. ISBN: 978-3-662-66769-9 (cit. on pp. 8, 18).

[17] A. a. r. Abu Ebayyeh and A. Mousavi. "A Review and Analysis of Automatic Optical Inspection and Quality Monitoring Methods in Electronics Industry". In: *IEEE Access* 8 (2020-10), pp. 183192–183271. DOI: 10.1109/ACCESS.2020.3029127 (cit. on pp. 8, 10–12, 14, 18).

[18] S. Ravikumar, K. Ramachandran, and V. Sugumaran. "Machine learning approach for automated visual inspection of machine components". In: *Expert Systems with Applications* 38.4 (2011), pp. 3260–3266. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2010.09.012. URL: https://www.sciencedirect.com/science/article/pii/S095741741000919X (cit. on pp. 8, 10).

[19] A. Corallo, V. Del Vecchio, and A. Di Prizio. "Advanced AI-Based Solutions for Visual Inspection: A Systematic Literature Review". In: *Proceedings of the 26th International Conference on Enterprise Information Systems (ICEIS 2024) - Volume 1*. Paper published under CC BY-NC-ND 4.0. SCITEPRESS – Science and Technology Publications, Lda, 2024, pp. 656–664. DOI: 10.5220/0012618000003690 (cit. on pp. 8, 21).

[20] X. Fang et al. "Research Progress of Automated Visual Surface Defect Detection for Industrial Metal Planar Materials". In: *Sensors* 20.18 (2020). ISSN: 1424-8220. DOI: 10.3390/s20185136. URL: https://www.mdpi.com/1424-8220/20/18/5136 (cit. on pp. 12, 21).

[21] K.-H. Tseng et al. "Artificial Intelligence based Vision Inspection for Manufacturing Industries". In: *Proceedings of the International Symposium on Smart Systems and Sensors (IS3C)*. 2023-06, pp. 346–348. DOI: 10.1109/IS3C57901.2023.00099 (cit. on p. 13).

[22] K. He et al. *Deep Residual Learning for Image Recognition*. 2015. arXiv: 1512.03385 [cs.CV]. URL: https://arxiv.org/abs/1512.03385 (cit. on pp. 13, 41, 42).

[23] J. Redmon et al. *You Only Look Once: Unified, Real-Time Object Detection*. 2016. arXiv: 1506.02640 [cs.CV]. URL: https://arxiv.org/abs/1506.02640 (cit. on pp. 14, 32, 38).

[24] A. Dosovitskiy et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 2021. arXiv: 2010.11929 [cs.CV]. URL: https://arxiv.org/abs/2010.11929 (cit. on pp. 15, 42).

[25] Z. Liu et al. *Swin Transformer: Hierarchical Vision Transformer using Shifted Windows*. 2021. arXiv: 2103.14030 [cs.CV]. URL: https://arxiv.org/abs/2103.14030 (cit. on pp. 15, 42).

[26] Z. Liu et al. *A ConvNet for the 2020s*. 2022. arXiv: 2201.03545 [cs.CV]. URL: https://arxiv.org/abs/2201.03545 (cit. on pp. 15, 42).

[27] E. Resendiz, J. M. Hart, and N. Ahuja. "Automated Visual Inspection of Railroad Tracks". In: *IEEE Transactions on Intelligent Transportation Systems* 14.2 (2013), pp. 751–760. DOI: 10.1109/TITS.2012.2236555 (cit. on p. 16).

[28] L. Leyendecker et al. "A Study on Data Augmentation Techniques for Visual Defect Detection in Manufacturing". In: Springer, 2023-03, pp. 73–94. ISBN: 978-3-662-66768-2. DOI: 10.1007/978-3-662-66769-9_6 (cit. on pp. 16–18).

[29] H. C. Nguyen and B. R. Lee. "Laser-Vision-Based Quality Inspection System for Small-Bead Laser Welding". In: *International Journal of Precision Engineering and Manufacturing* 15 (2014), pp. 415–423. DOI: 10.1007/s12541-014-0352-7 (cit. on p. 19).

[30] F. Sikstrom et al. "3D-Scanning for Weld Distortion Measuring". In: *2006 IEEE Instrumentation and Measurement Technology Conference Proceedings*. 2006, pp. 2132–2137. DOI: 10.1109/IMTC.2006.328524 (cit. on p. 19).

[31] M. Ivanov, A. Ulanov, and N. Cherkasov. "Visual Control of Weld Defects Using Computer Vision System on FANUC Robot". In: *2022 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM)*. 2022, pp. 859–863. DOI: 10.1109/ICIEAM54945.2022.9787245 (cit. on pp. 19, 20).

[32] P. Baldez et al. "A Passive Vision System for Weld Bead Inspection System Textures in Underwater Environments". In: *2020 Latin American Robotics Symposium (LARS), 2020 Brazilian Symposium on Robotics (SBR) and 2020 Workshop on Robotics in Education (WRE)*. 2020, pp. 1–6. DOI: 10.1109/LARS/SBR/WRE51543.2020.9307093 (cit. on pp. 19, 20).

[33] I. Mohd Shah et al. "Vision based Identification and Classification of Weld Defects in Welding Environments: A Review". In: *Indian Journal of Science and Technology* 9 (2016-05), pp. 1–15. DOI: 10.17485/ijst/2016/v9i20/82779 (cit. on p. 19).

[34] Y. Liu et al. "NDT Method for Weld Defects Based on FMPVit Transformer Model". In: *IEEE Access* 11 (2023), pp. 61390–61400. DOI: 10.1109/ACCESS.2023.3283589 (cit. on p. 20).

[35] *COCO: Common Objects in Context Dataset*. URL: https://cocodataset.org/#overview (visited on 2025-09-19) (cit. on p. 35).

A

Dataset Composition

This appendix provides a detailed breakdown of the number of images captured for each defect type across different positions during the data collection process. Tables A.1 and A.2 summarize the counts for the first and second tests, respectively, highlighting the distribution of defect instances.

Table A.1: Total number of images per defect per position (first test)

Defect/Position	Bottom 1	Top 1	Bottom 2	Top 2	Bottom 3	Top 3	Total
None	132	150	150	150	159	159	1035
Lack of Fusion	3	6	3	0	0	0	12
Porosity	0	0	0	3	0	0	3
Bead Height	6	0	3	6	0	0	12
Other Defects	18	3	3	0	0	0	24
Weld Overlay	0	0	0	0	0	0	0
Welding Reduction	0	0	0	0	0	0	0
Spatter	0	0	0	0	0	0	0
Missing Component	0	0	0	0	0	0	0
Total	159	159	159	159	159	159	1086

Table A.2: Total number of images per defect per position (second test)

Defect/Position	Bottom 2	Top 2	Total
None	78	117	195
Lack of Fusion	3	3	6
Porosity	0	0	0
Bead Height	0	0	0
Other Defects	0	0	0
Weld Overlay	18	3	21
Welding Reduction	0	0	0
Spatter	0	0	0
Missing Component	0	0	0
Total	99	123	222

B

Detailed Classification Results

This appendix reports the full per-class F1-scores and support values for all models and augmentation strategies, complementing the macro-F1 summaries provided in Chapter 6.3. The tables include the following columns:

- **Model** – the neural network architecture used for defect classification (e.g., ResNet18, ResNet50, Swin-T, ViT-Tiny, ConvNeXt-S);
- **Strat.** – the data augmentation strategy applied, numbered 1–4 as described in Chapter 6.3;
- **Macro-F1** – the macro-averaged F1-score across all defect classes, giving equal weight to each class;
- **Bead, Lack, None, Porosity, Overlay** – the defect classes: Bead Height, Lack of Fusion, None, Porosity, and Weld Overlay, respectively;
- **Bead-F1, Lack-F1, None-F1, Porosity-F1, Overlay-F1** – the per-class F1-scores for each defect type;
- **Bead-Sup, Lack-Sup, None-Sup, Porosity-Sup, Overlay-Sup** – the number of test samples (support) in each corresponding class.

Together, these per-class F1-scores and support values provide detailed insight into model performance for each defect category.

Table B.1: Original dataset: Macro-F1 and first few per-class F1-scores.

Model	Strat.	Macro-F1	Bead-F1	Bead-Sup	Lack-F1	Lack-Sup
ResNet18	1	0.694	0.281	8	0.594	8
	2	0.588	0.167	8	0.609	8
	3	0.330	0.000	9	0.279	9
	4	0.447	0.154	9	0.321	9
ResNet50	1	0.737	0.616	8	0.641	8
	2	0.388	0.242	8	0.144	8
	3	0.182	0.103	9	0.000	9
	4	0.179	0.000	9	0.067	9
Swin-T	1	0.746	0.500	8	0.620	8
	2	0.410	0.000	8	0.133	8
	3	0.175	0.000	9	0.000	9
	4	0.234	0.125	9	0.000	9
ViT-Tiny	1	0.613	0.603	8	0.199	8
	2	0.451	0.205	8	0.000	8
	3	0.201	0.000	9	0.000	9
	4	0.471	0.571	9	0.230	9
ConvNeXt-S	1	0.651	0.356	8	0.681	8
	2	0.310	0.074	8	0.626	8
	3	0.247	0.000	9	0.067	9
	4	0.363	0.133	9	0.167	9

Table B.2: Original dataset: remaining per-class F1-scores and support.

Model	Strat.	None-F1	None-Sup	Porosity-F1	Porosity-Sup	Overlay-F1	Overlay-Sup
ResNet18	1	0.829	39	1.000	1	0.769	9
	2	0.666	39	1.000	1	0.500	9
	3	0.680	38	0.348	0	0.333	9
	4	0.793	38	0.444	0	0.600	9
ResNet50	1	0.847	39	1.000	1	0.581	9
	2	0.616	39	0.667	1	0.273	9
	3	0.745	38	0.000	0	0.000	9
	4	0.700	38	0.000	0	0.000	9
Swin-T	1	0.847	39	1.000	1	0.764	9
	2	0.762	39	1.000	1	0.154	9
	3	0.750	38	0.000	0	0.000	9
	4	0.730	38	0.133	0	0.000	9
ViT-Tiny	1	0.730	39	0.889	1	0.644	9
	2	0.758	39	1.000	1	0.294	9
	3	0.720	38	0.000	0	0.500	9
	4	0.663	38	0.348	0	0.700	9
ConvNeXt-S	1	0.803	39	1.000	1	0.413	9
	2	0.783	39	0.000	1	0.067	9
	3	0.768	38	0.205	0	0.000	9
	4	0.790	38	0.250	0	0.632	9

Table B.3: Cropped dataset: macro-F1 and first few per-class F1-scores.

Model	Strat.	Macro-F1	Bead-F1	Bead-Sup	Lack-F1	Lack-Sup
ResNet18	1	0.891	0.767	8	0.852	8
	2	0.792	0.827	8	0.761	8
	3	0.465	0.282	9	0.267	9
	4	0.528	0.381	9	0.259	9
ResNet50	1	0.974	0.963	8	0.930	8
	2	0.848	0.919	8	0.582	8
	3	0.164	0.000	9	0.000	9
	4	0.503	0.534	9	0.480	9
Swin-T	1	0.974	0.963	8	0.930	8
	2	0.980	0.941	8	0.978	8
	3	0.536	0.308	9	0.444	9
	4	0.614	0.266	9	0.588	9
ViT-Tiny	1	0.343	0.067	8	0.157	8
	2	0.369	0.426	8	0.200	8
	3	0.209	0.205	9	0.000	9
	4	0.458	0.417	9	0.000	9
ConvNeXt-S	1	0.983	0.982	8	0.958	8
	2	0.959	0.960	8	0.911	8
	3	0.518	0.400	9	0.321	9
	4	0.646	0.348	9	0.371	9

Table B.4: Cropped dataset: remaining per-class F1-scores and support.

Model	Strat.	None-F1	None-Sup	Porosity-F1	Porosity-Sup	Overlay-F1	Overlay-Sup
ResNet18	1	0.942	39	1.000	1	0.892	9
	2	0.883	39	0.667	1	0.822	9
	3	0.771	41	0.455	0	0.500	9
	4	0.871	41	0.235	0	0.750	9
ResNet50	1	0.979	39	1.000	1	1.000	9
	2	0.907	39	1.000	1	0.830	9
	3	0.759	41	0.000	0	0.000	9
	4	0.914	41	0.167	0	0.857	9
Swin-T	1	0.979	39	1.000	1	1.000	9
	2	0.983	39	1.000	1	1.000	9
	3	0.929	41	0.286	0	0.853	9
	4	0.945	41	0.250	0	0.891	9
ViT-Tiny	1	0.855	39	0.000	1	0.637	9
	2	0.815	39	0.000	1	0.405	9
	3	0.777	41	0.000	0	0.000	9
	4	0.871	41	0.467	0	0.522	9
ConvNeXt-S	1	0.992	39	1.000	1	0.980	9
	2	0.991	39	1.000	1	0.933	1
	3	0.934	41	0.286	0	0.757	9
	4	0.930	41	0.231	0	0.888	9