

THERMAL IMAGING: DETECTING DEFECTS WITH SEMANTIC SEGMENTATION

Submitted in partial fulfilment of the requirements for the award of
Bachelor of Engineering degree in Mechatronics

By

KEVIN RAJADURAI E (Reg. No. 40090015)

CHIDAMBARESH N (Reg. No. 40090703)



**DEPARTMENT OF MECHATRONICS
SCHOOL OF MECHANICAL ENGINEERING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with grade “A++” by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600119

APRIL - 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

DEPARTMENT OF MECHATRONICS

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of KEVIN RAJADURAI E (Reg. No. 40090015) and CHIDAMBARESH N (Reg. No. 40090703) who carried out the project entitled "**THERMAL IMAGING: DETECTING DEFECTS WITH SEMANTIC SEGMENTATION**" under our supervision from December 2023 to April 2024.

Internal Guide

Dr. J.R. DEEPAK., M.E., Ph.D.,

Head of the Department

Dr. B. KANIMOZHI., M.E., Ph.D.,

Submitted for Viva voce Examination held on 05-04-2024

Internal Examiner

External Examiner

DECLARATION

We KEVIN RAJADURAI E (40090015) and CHIDAMBARESH N (40090703) hereby declare that the Project Report entitled **"THERMAL IMAGING : DETECTING DEFECTS WITH SEMANTIC SEGMENTATION"** done by us under the guidance of **Dr. J. R. DEEPAK, M.E., Ph.D.**, is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in Mechatronics.

1.

Kevin

2.

Chidambaresh

DATE: 05.04.2024

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to Board of Management of **Sathyabama institute of science and technology** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey our sincere thanks to **Dr. B. KANIMOZHI., M.E., Ph.D.**, Head of the Department, Department of Mechatronics for providing us necessary support during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to our Project Guide **Dr. J. R. DEEPAK., M.E., Ph.D.**, for his valuable guidance, suggestions, and constant encouragement paved way for the successful completion of our project work.

We wish to express our thanks to all Teaching and Non-teaching staff members of the Department of Mechatronics who were helpful in many ways for the completion of the project.

We also specially thank our parents for their continuous support and cooperation throughout the end for the completion of the project.

ABSTRACT

Thermal imaging systems are high-quality system known for non-invasive, high-resolution defect identification. The infrared radiation emissions are captured by these thermal imaging systems to carry out meticulous inspections without any physical contact. The thermal images captured is of high contrast, in a color scale which maps the heat signatures of the object that is inspected, thereby differentiating the temperature variations with respect to that color scale. But these thermal images generated in the form of RGB images do not indicate accurate temperature values. To examine these temperature variations, addition of deep learning algorithm a subfield of machine learning, such as semantic segmentation assistance to categorize these thermal images into categories in the pixel level. YOLOv8 is a deep learning module which is designed for real time object segmentation, making them suitable for applications that require quick and accurate identification of features of the image. When integrated with thermal imaging systems, these YOLOv8 modules create a real breakthrough in the field of Non-Destructive Testing (NDT). In this research work, a thermal imaging defect detection system with semantic segmentation is developed to detect various defects in the thermal images of the welded metal sheets. The thermal imaging defect detection system includes an induction coil to preheat the welded metal sheet, Lepton 3.1R camera module to generate thermal image and a Raspberry Pi 4 for transmitting the image to the computer in which the YOLOv8 module is executed. Rob-flow software pre-processes and annotate the thermal image into labels. These labeled images are categorized and subsequently fed into the YOLOv8 module for model training which enables the system to detect the labeled defects. The fusion of thermal imaging with deep learning such as the YOLOv8 model in this research, elevates the field of inspection (NDT) by enabling automated, high-speed analysis of vast thermal datasets which enhances defect detection accuracy and its utility in the field of non-destructive testing (NDT).

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE No
	ABSTRACT	V
	LIST OF FIGURES	viii
	LIST OF TABLES	ix
	LIST OF SYMBOLS AND ABBREVIATIONS	x
1	INTRODUCTION	
	1.1 Welding	1
	1.2 Weld Defects	1
	1.2.1 Incomplete Penetration	1
	1.2.2 Incomplete Fusion	2
	1.2.3 Porosity	3
	1.2.4 Cracks	3
	1.2.5 Undercut	4
	1.2.6 Overlap	5
	1.2.7 Spatter	5
	1.2.8 Slag Inclusions	6
	1.3 Non-Destructive Testing	7
	1.4 Types of NDT Techniques	8
	1.5. Thermography	9
	1.5.1 Types of Thermography	9
	1.5.2 Advantages of Thermography Systems	11
	1.5.3 Disadvantages of Thermography Systems	11
	1.6.1 Importance of Computer Vision	12
	1.6.1 Classifications of Computer Vision	12
	1.6.2 Use of Computer Vision In Thermography	13
	1.6.3 Applications of Computer Vision Systems	14
	1.6.4 Challenges and Limitations faced when implementing Computer vision	14
	1.6.5 Classifications of Computer Vision	14
	1.7 Use of Computer Vision in Thermography	16
	1.8 Introduction to Deep Learning	18

	1.8.1 Convolutional Neural Networks	18
	1.8.2 Key Components of Convolutional Neural Networks:	19
	1.9 Image Segmentation	20
	1.9.1 Applications of Image Segmentation:	20
	1.9.2 Various Image Segmentation techniques	21
	1.9.3 Advantages of Semantic Segmentation	22
	1.10 Object Detection Systems	25
	1.10.1 Requirements for Object Detection systems	26
	1.10.2 Applications in detecting Weld Defects in Thermal Images:	27
	1.10.3 Types of Object Detection	28
	1.10.4 One step And Two step networks	29
	1.10.5 Object Detection algorithms	31
2	LITERATURE SURVEY	34
	2.1 Literature Survey based on Thermal Imaging Systems	34
	2.2 Literature Survey based on YOLO models	36
3	AIM AND SCOPE	38
	3.1 Aim	38
	3.2 Objective	38
	3.3 Scope of the Investigation	38
4	EXPERIMENTAL SETUP AND METHODS	40
	4.1 Overall process flow of thermal imaging-based defect detection system	40
	4.2 Experimental Setup For Thermal Imaging System	41
	4.3 Working Of The Thermal Imaging System	42
	4.4 Data Preparation and Augmentation of Thermal Images	46
	4.5 Annotation Of Images Using Roboflow	47
	4.6 YOLO (OBJECT DETECTION ALGORITHM)	48
	4.6.1 Key steps involved in the workflow of YOLO	49
	4.6.2 Architecture of YOLO	50
	4.6.3 Training of deep learning model using YOLO v8	51
5	RESULTS AND DISCUSSION	53
	5.1 Training Results of the Defect Detection Model	55

	5.1.1 Confusion Matrix of the Model	55
	5.1.2 Model Evaluation based on Loss Functions	57
	5.1.3 Model Evaluation based on Metrics	58
	5.2 Accuracy of the Predicted Images of the Model	59
	5.3 Predicting images in real time	59
6	SUMMARY AND CONCLUSION	63
	REFERENCES	65
	ANNEXURE	70

LIST OF FIGURES

FIGURE NO	TITLE	PAGE No.
1.1	Lack of Fusion in weld bead	2
1.2	Porosity in weld bead	3
1.3	Crack in weld bead	4
1.4	Undercut of weld bead	5
1.5	Overlap on weld bead	6
1.6	Slag Inclusion in weld bead	7
1.7	Types of NDT Techniques	8
1.8	Types of Thermography Techniques	9
1.9	Types of Computer Vision	15
1.10	Types of Image Segmentation Techniques	21
1.11	Two Step Networks	29
1.12	One Step Networks	30
1.13	Evolution of Object Detection Algorithms	31
4.1	Proposed Defect Detection System	40
4.2	Circuit Connection Diagram of the thermal imaging setup	42
4.3	Matrix generated on the Raspberry Pi denoting that the Lepton 3.1R is detected.	43
4.4	Lines generated by Raspberry Pi using rpi_vsync_app	44
4.5	Thermal image obtained using the Thermal Imaging Setup	45
4.6	Images pre-processed and augmented in Roboflow	46
4.7	Images Annotated in Roboflow for their respective defects	47
4.8	Code snippet from Roboflow used in Google Collab	47
4.9	Completion of training of model	52
5.1	Confusion matrix	56
5.2	Training and Validation Losses of the model	57
5.3	Metrics of the model bases on Bounding Boxes and Maps	58
5.4	Thermal images of weld predicted by the YOLO v8 model	59
5.5	Python program created using the YOLO model	61

5.6	Image opened using the python program	62
5.7	Prediction made on the image using the python program	62

LIST OF TABLES

TABLE NO	TITLE	PAGE NO
1.1	Key features of the YOLO models	33
2.1	Categorization based on Material Application, Camera used, Implemented models	34
4.1	Specification of Raspberry PI 4	42
4.2	Distribution of images	45
5.1	Training Results for each Epoch	54
5.2	Values obtained from the Confusion Matrix	57

LIST OF SYMBOLS AND ABBREVIATIONS

SMAW	Shielded Metal Arc Welding
GMAW	Gas Metal Arc Welding
GTAW	Gas Tungsten Arc Welding
SAW	Submerged Arc Welding
FCAW	Flux-Cored Arc Welding
PAW	Plasma Arc Welding
EBW	Electron Beam Welding
HAZ	Heat Affected Zone
NDT	Non-destructive testing
NDT	Non-destructive techniques
VI	Visual Inspection
RT	Radiographic Testing
UT	Ultrasonic Testing
MT	Magnetic Particle Testing
PT	Liquid Penetrant Testing
ET	Eddy Current Testing
AE	Acoustic Emission Testing
IRT	Infrared Thermography
IR	infrared
DL	Deep Learning
ML	Machine Learning
CNNs	Convolutional Neural Networks
OCR	optical character recognition
PCs	personal computers
RNNs	Recurrent Neural Networks
NLP	Natural language processing
GANS	Generative Adversarial Networks
FNNs	Feedforward Neural Networks

GRU	Gated Recurrent Unit
ReLU	Rectified Linear Unit
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
VGG	Visual Geometry Group
SGD	Stochastic Gradient Descent
FCNs	Fully Convolutional Networks
SVMs	Support Vector Machines
NMS	non-maximum suppression
HOG	Histogram of Oriented Gradients
LBP	Local Binary Patterns
SVMs	Support Vector Machines
CRFs	Conditional Random Fields
RPN	region proposal network
YOLO	You Only Look Once
SSD	Single Shot MultiBox Detector
R-CNN	Region-based Convolutional Neural Network
MRI	Magnetic resonance imaging
CT	Computed tomography
mAP	mean average precision
COCO	Common objects in contexts
FLIR	Forward looking infrared
E-ELAN	Extended efficient layer aggregation network
KRR	Kernel Ridge Regression
IRMV	Infrared machine vision
DCNN	deep learning Convolutional Neural Networks
CFRP	Carbon Fiber Reinforced polymer
SBC	single-board computer
HDMI	High-Definition multimedia Interface
CPU	Central Processing Unit
RAM	Random access memory
GPU	Graphics Processing unit
GPIO	General Purpose Input/Output

OS	Operating system
SOTA	State-Of-The-Art
P	Precision
R	Recall
IoU	Intersection over Union
TP	True Positives
TN	True Negatives

CHAPTER 1

INTRODUCTION

1.1 WELDING

Welding has been integral to every industry globally and serves as the backbone in all major aspects like in aerospace, automotive, construction and energy industries. Welding, a manufacturing process that involves fusion of metals, by melting and joining them. This is achieved by applying heat to the materials to be joined, often with the addition of a material called filler, resulting in a strong and durable connection once the melted material solidifies. Welding is commonly used in construction, fabrication, and various industries to create permanent bonds between metal components, forming structures and products. Different welding techniques exist, such as TIG welding, MIG welding, Arc welding, and more, each suited to specific applications based on factors like material type, thickness, and desired joint characteristics. Timely rectification of defects not only reduces expenses related to rework, material wastage, and downtime but also maintains high-quality standards. To prevent these the welded components are inspected using various Non-destructive techniques.

1.2 TYPES OF WELDING DEFECTS

Weld defects are undesired irregularities or imperfections that can occur during the welding process, compromising the quality and integrity of the welded joint. Various factors, such as welding technique, material properties, and environmental conditions, can contribute to the formation of weld defects. Understanding the causes of weld defects is essential for preventing their occurrence and ensuring high-quality welds.

1.2.1 Incomplete Penetration:

Incomplete penetration is a weld defect that occurs when the weld metal does not fully penetrate the joint, leading to an insufficient bond between the pieces being joined. This defect weakens the structural integrity of the weld and can compromise the overall strength of the joint. This defect has a characteristic visible gap or unfilled space is

present in the joint where the weld metal failed to penetrate completely. The weld metal may not fuse completely with the base material, resulting in an incomplete bond. Some causes for incomplete penetration are due to the following reasons. If the welding process does not generate enough heat, the molten weld metal may not reach the depth required for full penetration. Incorrect welding angles, travel speed, or electrode positioning can contribute to incomplete penetration. Poor joint design, such as inadequate bevel angles or insufficient groove depth, can hinder proper penetration.

1.2.2 Lack of Fusion:

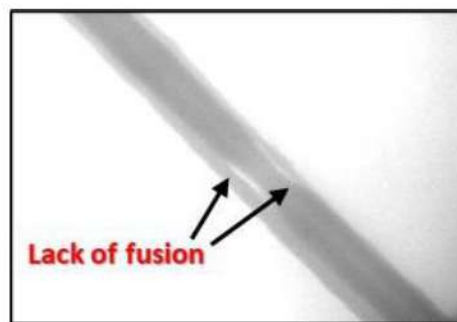


Figure 1.1: Lack of Fusion in weld bead

Lack of fusion is a weld defect that occurs when the weld metal does not fully fuse with the base material, resulting in a weak joint. In this defect, there is a lack of metallurgical bonding between the weld and the base metal, compromising the structural integrity of the weld. Some characteristic feature like a visible seam or weak joint is present. A visible line or seam is present between the weld metal and the base material, indicating a lack of fusion as seen in Figure 1.1. The weld may not effectively join the pieces together, leading to reduced strength and potential failure under stress. Incomplete fusion can be caused due to, insufficient generation of heat due to which the molten weld metal may not fuse completely.

1.2.3 Porosity

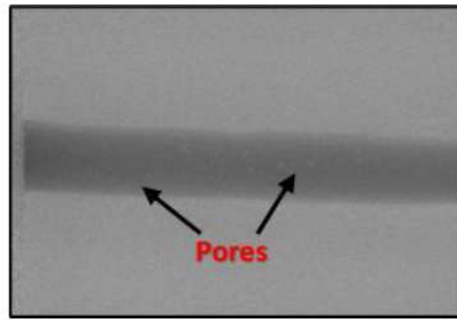


Figure 1.2: Porosity in weld bead

Porosity is a weld defect characterized by the presence of small cavities or bubbles in the weld metal, which can reduce the strength and integrity of the weld, potentially leading to structural issues. The key characteristics of porosity include the appearance of bubble-like holes as seen in Figure 1.2 , taking the form of small, spherical, or elongated cavities within the weld metal. Improper shielding gas, inadequate or incorrect flow during welding, can contribute to gas entrapment, resulting in porosity. Furthermore, incorrect welding parameters, such as inappropriate current, voltage, or travel speed settings, can lead to porosity issues. Preventing and addressing porosity involves ensuring that workpieces, electrodes, and filler materials are dry and free from moisture is crucial.

1.2.4 Cracks

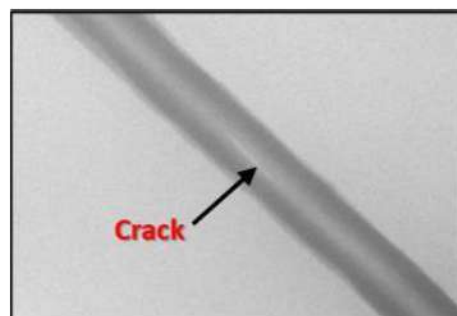


Figure 1.3: Crack in weld bead

Cracks are a serious weld defect characterized by unintended fissures, as seen in Figure 1.3, in the welded material. These fissures can compromise the structural integrity of the weld, leading to potential failure under stress. Cracks can vary in size and shape, and they may be visible on the surface of the weld or extend into the interior of the welded joint. The causes of cracks in welding are diverse and may include

factors such as rapid cooling, high levels of stress during or after welding, and inadequate preparation of the materials being joined. The types of cracks commonly encountered in welding include: Hot Cracks which are formed during the solidification process when the metal is still in a semi-liquid state. Cold Cracks occur after the weld has solidified and cooled. Stress Cracks develop over time due to repeated loading or external stresses. These are mainly caused due to ongoing stress, often related to the service conditions of the welded structure.

1.2.5 Undercut

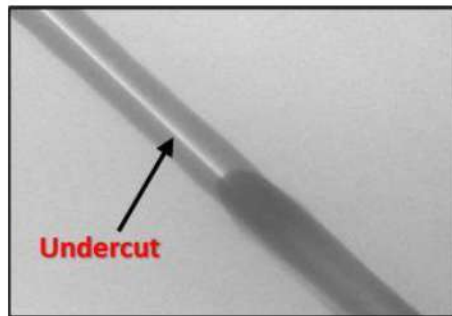


Figure 1.4: Undercut of weld bead

Undercut is a weld defect characterized by the erosion or groove formation along the edges of the weld, resulting in a depression or notch in the base metal adjacent to the weld bead. This defect weakens the joint, reducing its cross-sectional area and potentially compromising its structural integrity. The key characteristics of undercutting include a visible groove or depression along the edges of the weld bead as seen in Figure 1.4. Additionally, the metal at the edge of the weld is thinned or removed, leading to a reduction in the cross-sectional area. Several factors can contribute to undercutting, such as excessive welding current, incorrect electrode angle (too steep or too shallow), incorrect travel speed (too slow or too fast), and inadequate welder skill or experience in maintaining proper welding techniques. Preventing and addressing undercutting involves various measures. Optimizing welding parameters, including current, voltage, and travel speed, is crucial to prevent excessive heat and metal removal. Ensuring the correct angle of the electrode helps minimize excessive melting of the base metal. Controlling the travel speed during welding is essential to avoid overexposure and excessive melting.

1.2.6 Overlap

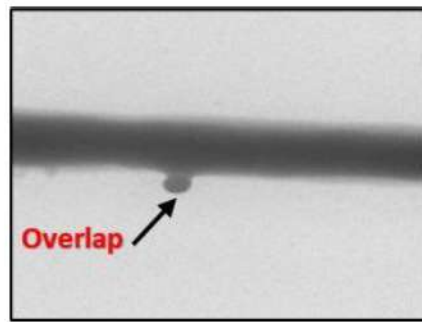


Figure 1.5: Overlap on weld bead

Overlap is a weld defect characterized by the excessive build-up of weld metal, creating a raised surface that extends beyond the desired boundary of the weld joint. This defect can compromise the structural integrity of the weld, leading to issues such as reduced cross-sectional area and potential stress concentration points. The key characteristics of overlap include excess weld metal, where the weld metal extends beyond the intended weld joint, forming a protruding surface with a raised bead as seen in Figure 1.5 . Several factors contribute to the occurrence of overlap. Incorrect welding parameters, such as inappropriate settings for welding current, voltage, or travel speed, can result in excessive deposition of weld metal. Inadequate welding technique, including poor control of the welding gun, incorrect manipulation of the electrode, or improper weaving patterns, can also contribute to overlap. Additionally, using an excessive amount of filler material, especially in processes where filler material is added, can lead to overlap. Preventing and addressing overlap involves several measures. Optimizing welding parameters, including current, voltage, and travel speed, is crucial to achieving the desired deposition without excessive build-up. Ensuring improved welding technique, such as proper control of the welding gun, precise manipulation of the electrode, and correct weaving patterns during welding, helps avoid overlap. Appropriate usage of filler material, following recommended amounts to avoid excessive build-up, is also important.

1.2.7 Spatter:

Spatter is a weld defect characterized by the expulsion of small droplets of molten metal from the welding process. During welding, small droplets of molten metal are expelled and may land on adjacent surfaces, resulting in a messy and uneven appearance on the surface of the weld and surrounding areas. While spatter itself may

not directly compromise the integrity of the weld, it can lead to surface irregularities, cosmetic issues, and, in some cases, affect the surrounding environment. Several factors contribute to the occurrence of spatter such as inappropriate settings for welding current, voltage, or travel speed. Inadequate or improper shielding gas during welding is another factor that can contribute to spatter. Additionally, the wire stick-out length (the length of the wire extending from the contact tip) and the type of electrode used can impact the amount of spatter generated. While spatter is primarily a cosmetic concern, addressing it is important to avoid wasted materials, increased cleanup efforts, and to maintain the overall aesthetics of the welded joint. Preventing and addressing spatter involves several measures. Optimizing welding parameters, including current, voltage, and travel speed, is crucial to minimize spatter. Using the correct shielding gas and ensuring proper flow rates help reduce spatter. Maintaining the proper wire stick-out length and selecting an electrode type that minimizes spatter are also essential considerations. Regular visual inspections during and after the welding process, as part of quality control practices, help identify signs of spatter, enabling corrective actions to ensure a clean, well-finished weld.

1.2.8 Slag Inclusions

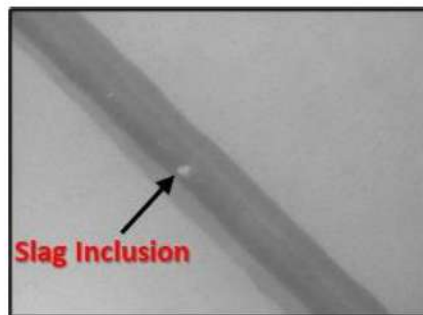


Figure 1.6: Slag Inclusion in weld bead

Slag inclusions are a weld defect characterized by the presence of foreign materials trapped in the weld metal. These foreign materials, often non-metallic particles or substances like slag, can compromise the mechanical properties of the weld, leading to reduced strength and potential failure under stress. Inclusions may be visible on the surface of the weld or may exist within the weld itself. The key characteristics of inclusions include the presence of non-metallic particles, such as slag or other contaminants, within the weld metal. These inclusions may appear as irregularities on the surface of the weld or may be hidden within the weld as seen in Figure 1.6. Several

factors contribute to the occurrence of inclusions. Poor welding techniques, including inadequate cleaning or improper manipulation of the electrode, can introduce contaminants into the weld. In processes involving flux, inadequate coverage may lead to incomplete removal of contaminants, resulting in inclusions. Additionally, inappropriate welding parameters, such as excessive heat or incorrect travel speed, may contribute to the entrapment of foreign materials.

1.3 NON-DESTRUCTIVE TESTING

Non-destructive testing (NDT) encompasses a range of methods employed to assess the properties, integrity, and quality of materials, components, or structures without inflicting harm upon the tested object. NDT holds significant importance in guaranteeing the safety, dependability, and efficiency of diverse engineering applications across sectors including aerospace, automotive, construction, manufacturing, oil and gas, and infrastructure. NDT techniques are based on physical principles such as electromagnetism, acoustics, optics, radiation, and mechanical properties. These techniques rely on the interaction between the test specimen and a probing signal (e.g., sound waves, electromagnetic radiation) to detect and analyze anomalies, defects, or discontinuities in the material. NDT majorly allows for the evaluation of materials and structures without damaging or altering the test specimen. It also provides valuable information about the integrity, quality, and performance of materials, components, or structures. Not only does it help in inspections but also helps to identify defects, anomalies, or discontinuities early in the manufacturing process, reducing the risk of costly failures or accidents. NDT allows us to periodically inspect critical components or structures to ensure continued safety and reliability over their service life.

NDT finds its application in various fields such as in aerospace for Inspection of aircraft components, engines, and structures. In automotive for Quality control of automotive parts and assemblies. In Oil and Gas for Inspection of pipelines, pressure vessels, and offshore structures. In construction for evaluation of concrete, steel, and composite structures. In manufacturing for Quality assurance of welds, castings, and machined components. In infrastructure for assessment of bridges, dams, tunnels, and buildings.

1.4 TYPES OF NDT TECHNIQUES

Non-destructive techniques due to their non-invasive nature are preferred for identifying weld defects because they preserve the welded component. They usually are cost-effective and efficient, and versatile and provide flexible usability by allowing the detection of both surface and subsurface defects. There are various techniques used for detecting defects, Figure 1.7 focuses on detecting defects for weld defects

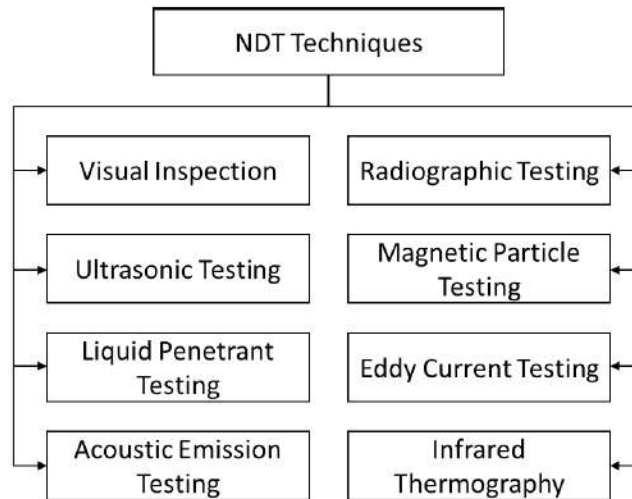


Figure 1.7: Types of NDT Techniques

- Visual Inspection (VI): VI, the most fundamental type of NDT, involves visually scrutinizing welds for surface irregularities, discontinuities, cracks, or other imperfections.
- Radiographic Testing (RT): RT utilizes X-rays or gamma rays to generate images of the weld's internal structure, revealing defects like porosity, cracks, lack of fusion, or incomplete penetration.
- Ultrasonic Testing (UT): UT employs high-frequency sound waves (ultrasound) to examine the weld's internal structure, identifying defects such as cracks, lack of fusion, incomplete penetration, or porosity.
- Magnetic Particle Testing (MT): MT detects surface and near-surface defects in ferromagnetic materials by magnetizing the weld and applying ferromagnetic particles, which gather at defective areas like cracks or lack of fusion.
- Liquid Penetrant Testing (PT): PT detects surface-breaking defects by applying a liquid penetrant to the weld surface, which seeps into any surface cracks or openings.

After excess penetrant removal, a developer is applied to reveal defects.

- Eddy Current Testing (ET): ET identifies surface and near-surface defects in conductive materials by inducing eddy currents into the material and detecting changes in electrical conductivity caused by defects like cracks or lack of fusion.
- Acoustic Emission Testing (AE): AE monitors the acoustic emissions (ultrasonic waves) emitted by stressed materials, detecting defects such as cracks and lack of fusion during welding or post-welding processes.
- Infrared Thermography (IRT): IRT detects defects by measuring the heat emitted from the weld, with variations indicating issues like lack of fusion, incomplete penetration, or cracks.

1.5 THERMOGRAPHY

One of the most popular non-contact temperature measurement IR based imaging techniques is thermography, which uses heat patterns on a target material to find flaws. It became popular among NDT techniques in the relevant industry, mostly because its inspection time was relatively short, it didn't require touch or coupling like other NDT techniques, it was simple to comprehend the data because the findings were often photos or videos, and it had a shorter inspection process. As a result, it requires less knowledge and training, as well as no dangerous from radiation sources. It offers complete field information, is non-contact, and is rather quick. By exaggerating the fracture, this technique has previously been employed to find metal and concrete flaws by either vibrations or eddy currents. The strategy employed herefor igniting the Pulsed infrared (IR) radiation is used to detect surface fissures. The capacity or ability of an object to absorb incident radiated energy is always the same as the capacity to emit its own energy as radiation ($\epsilon = \alpha$) Due to multiple reflections of incident pulse radiation in a crack, a larger portion of the energy is absorbed compared to a single reflection at the surface.

1.5.1 Types of Thermography

There are several types of thermographic NDT techniques, each with its own approach and applications. Figure 1.8 lists out these types.

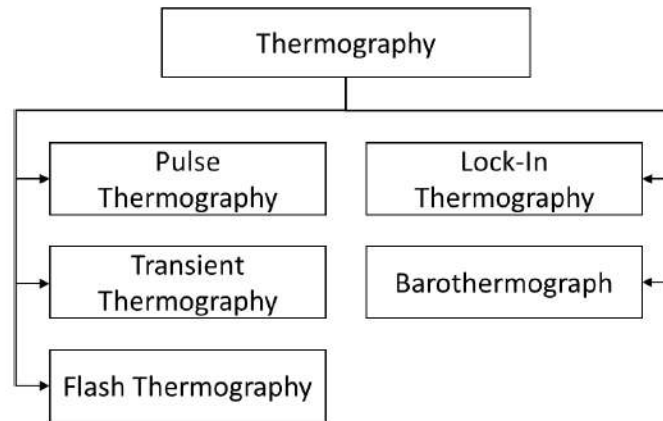


Figure 1.8: Types of Thermography Techniques

- **Pulse Thermography:** In pulse thermography, a heat pulse is applied to the surface of the material being tested using a flash lamp or other heat source. The surface temperature is then monitored using an infrared camera. Variations in temperature distribution over time can indicate the presence of defects such as delamination's, cracks, or voids within the material.
- **Lock-in Thermography:** Lock-in thermography involves modulating the heat source at a specific frequency and analyzing the phase and amplitude of the resulting thermal response. This technique is particularly effective for detecting subsurface defects in composite materials or layered structures.
- **Transient Thermography:** Transient thermography techniques involve analyzing the thermal response of a material to a short-duration heat pulse. By monitoring the cooling process, defects such as cracks or voids can be identified based on their influence on heat transfer characteristics.
- **Barothermograph:** Barothermograph combines thermal imaging with mechanical excitation. By applying mechanical vibrations to the material under inspection and monitoring the resulting thermal response, defects such as disbands or delamination's can be detected based on their impact on heat dissipation.
- **Flash Thermography:** Flash thermography uses a high-intensity flash lamp to rapidly heat the surface of the material. The subsequent thermal response is captured using an infrared camera. This technique is suitable for detecting surface and subsurface defects in a variety of materials, including metals, composites, and ceramics.

1.5.2 Advantages of Thermography Systems

- **Non-contact Inspection:** Thermography allows for non-contact inspection, which means there is no need to physically touch the weld or the material being inspected. This reduces the risk of damage to the weld and eliminates the need for surface preparation, making inspections faster and more convenient.
- **Rapid Inspection:** Thermographic inspections can be conducted rapidly compared to traditional NDT methods. With techniques such as pulse thermography or flash thermography, defects can be quickly identified by analyzing the thermal response of the weld, leading to faster overall inspection times.
- **Detection of Surface and Subsurface Defects:** Thermography is capable of detecting both surface and subsurface defects in welds. This includes cracks, voids, lack of fusion, porosity, and other anomalies that may be present within or beneath the surface of the weld. **Wide Area Coverage:** Thermal imaging cameras can capture a wide area of the weld in a single image, allowing inspectors to assess large sections of welds simultaneously. This facilitates comprehensive inspections and ensures that defects are not missed, even in complex or intricate weld geometries.
- **Quantitative Analysis:** Thermography can provide quantitative data on the size, depth, and location of defects within the weld. This information helps in assessing the severity of defects and determining their impact on the structural integrity of the weld.

1.5.3 Disadvantages of Thermography Systems:

- **Surface Condition Sensitivity:** Thermography is sensitive to variations in surface conditions such as roughness, oxidation, or coatings. Poor surface conditions can affect the accuracy of thermal imaging results and may require additional surface preparation to ensure reliable defect detection.
- **Limited Penetration Depth:** The depth of penetration of thermal energy into the weld material is limited, especially for techniques such as pulse thermography or flash thermography. This means that subsurface defects may not be detected effectively if they are located too deep within the weld.
- **Material Dependency:** The effectiveness of thermography for weld inspection can

vary depending on the type of material being inspected. Some materials may exhibit different thermal properties that can affect the detection of defects, requiring adjustments in inspection techniques or parameters.

- **Complex Data Interpretation:** Interpreting thermal images and identifying defects within welds can be complex and may require specialized training and expertise. False positives or false negatives can occur if the thermal response is misinterpreted, leading to potential inaccuracies in defect detection.
- **Equipment Cost:** Thermal imaging equipment can be expensive to purchase and maintain, especially high-resolution infrared cameras suitable for detailed weld inspections. Additionally, specialized software and hardware may be required for data analysis and interpretation, adding to the overall cost of implementing thermographic NDT techniques.

Despite these limitations, thermography remains a valuable tool for weld defect detection, offering fast, reliable, and non-destructive inspection capabilities for a wide range of industrial applications.

1.6. IMPORTANCE OF COMPUTER VISION

Computer vision is a multidisciplinary field that enables computers to interpret and understand visual information from the real world. It involves the development of algorithms, models, and systems that can extract meaningful insights from digital images or videos. Computer vision aims to replicate the human visual system's ability to perceive, analyze, and interpret visual data, allowing machines to perform tasks such as object detection, recognition, segmentation, tracking, and scene understanding.

1.6.1 Overview of Components of Computer Vision Systems

Image Acquisition: The process of capturing visual data from the real-world using cameras or other imaging devices. This data may be in the form of images (2D) or videos (3D). **Pre-processing:** Involves enhancing the quality of acquired images or videos to improve the performance of subsequent computer vision algorithms. This may include tasks such as noise reduction, image stabilization, color correction, and image scaling. **Feature Extraction:** Identifying relevant visual features within images or videos that can be used for further analysis. Features may include edges, corners,

textures, shapes, colors, or patterns. **Feature Representation:** Encoding extracted features into a suitable format for processing by machine learning algorithms. This may involve techniques such as feature descriptors, histograms, or deep feature embeddings. **Recognition and Classification:** Identifying and categorizing objects or patterns within images or videos based on their visual characteristics. This may involve techniques such as object detection, image classification, object recognition, or scene understanding. **Segmentation:** Partitioning images or videos into meaningful regions or objects based on similarities in visual properties. This may involve techniques such as semantic segmentation, instance segmentation, or edge detection. **Tracking:** Following the movement or changes of objects or regions of interest over time within a sequence of images or videos. This may involve techniques such as object tracking, motion estimation, or optical flow analysis. **Scene Understanding:** Inferring higher-level information about the environment or context depicted in images or videos. This may involve tasks such as depth estimation, 3D reconstruction, scene interpretation, or activity recognition.

1.6.2 Key Techniques and Approaches:

Traditional Computer Vision: Includes classical techniques such as image filtering, edge detection, corner detection, and template matching. These techniques are often based on handcrafted features and rules. **Machine Learning-based Approaches:** Utilizes machine learning algorithms to automatically learn patterns and features from data. Techniques such as supervised learning, unsupervised learning, and reinforcement learning are applied to tasks such as object recognition, classification, and segmentation. **Deep Learning:** A subset of machine learning that utilizes deep neural networks with multiple layers to learn complex representations of visual data. **Convolutional Neural Networks (CNNs)** are particularly well-suited for tasks such as image classification, object detection, and semantic segmentation. **Transfer Learning:** Leveraging pre-trained deep learning models on large datasets to solve computer vision tasks with limited training data. This approach allows for faster and more efficient development of computer vision systems.

1.6.3 Applications of Computer Vision Systems

Computer vision has a wide range of applications across various industries and domains, including Autonomous Vehicles Object detection, tracking, and scene understanding for self-driving cars and drones, Surveillance and Security: Intrusion detection, activity recognition, and facial recognition for monitoring and security systems, Manufacturing and Robotics: Quality inspection, object manipulation, and robotic vision for industrial automation, Agriculture: Crop monitoring, pest detection, and yield estimation using aerial and satellite imagery.

1.6.4 Challenges and Limitations Faced when implementing computer vision

- **Data Quality and Quantity:** Availability of labeled training data and the quality of annotations can significantly impact the performance of computer vision models.
- **Robustness to Variability:** Computer vision systems may struggle with variations in lighting conditions, occlusions, viewpoint changes, and background clutter.
- **Interpretability:** Deep learning models often lack interpretability, making it challenging to understand their decision-making process and reasoning.
- **Ethical and Privacy Concerns:** Issues such as bias in datasets, facial recognition privacy, and surveillance ethics raise important ethical considerations in computer vision applications.

Computer vision plays a crucial role in enabling machines to perceive and understand visual information from the real world, with applications spanning various industries and domains. Advances in machine learning and deep learning continue to drive innovation in this field, addressing challenges and unlocking new possibilities for real-world applications.

1.6.5 Classifications of Computer Vision

Computer vision systems can be classified into several categories based on their functionality, architecture, and application requirements. Figure 1.9 lists out some of these classifications of machine vision systems:

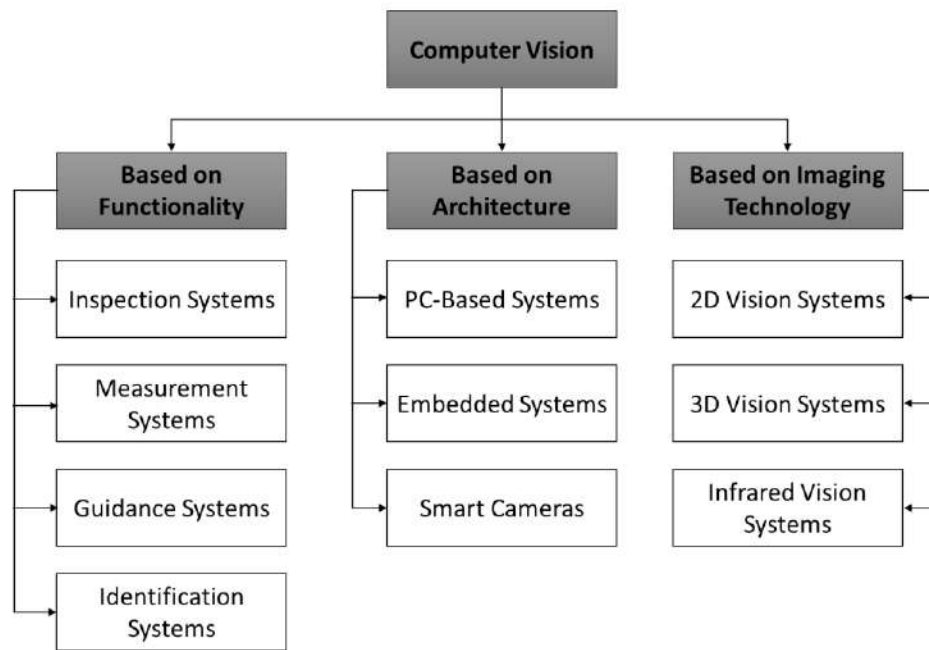


Figure 1.9: Types of Computer Vision

Based on Functionality the computer vision systems can be classified as

- **Inspection Systems:** These systems are designed to detect defects, anomalies, or irregularities in products or materials. They often involve tasks such as surface inspection, quality control, and defect detection.
- **Measurement Systems:** These systems are focused on accurately measuring dimensions, distances, angles, or other physical properties of objects. They are commonly used in metrology, dimensional analysis, and precision manufacturing applications.
- **Guidance Systems:** These systems provide real-time guidance or feedback to automated machinery or robots based on visual information. They are used in applications such as robotic navigation, assembly, and pick-and-place operations.
- **Identification Systems:** These systems are used to identify and recognize objects, patterns, or features within images or videos. They include tasks such as object recognition, barcode reading, facial recognition, and optical character recognition (OCR).

Based on Architecture the computer vision systems can be classified as

- **PC-Based Systems:** These systems utilize standard personal computers (PCs) equipped with specialized hardware such as frame grabbers and cameras. They are flexible and customizable, allowing for the development of custom vision algorithms and applications.
- **Embedded Systems:** These systems are designed for integration into industrial machines, equipment, or production lines. They often consist of compact, embedded hardware platforms with built-in processing capabilities and interfaces for connecting cameras and sensors.
- **Smart Cameras:** These are self-contained vision systems that integrate cameras, processors, and interfaces into a single device. Smart cameras are often used in applications where space is limited or where real-time processing is required at the edge of the network.

Based on Imaging Technology the computer vision systems can be classified as:

- **2D Vision Systems:** These systems capture and process images in two dimensions (2D), providing information about the spatial arrangement and features of objects within the field of view.
- **3D Vision Systems:** These systems capture and process images in three dimensions (3D), enabling depth perception and spatial understanding of objects. They are used in applications such as 3D reconstruction, surface profiling, and volume measurement.
- **Infrared (IR) Vision Systems:** These systems utilize infrared imaging technology to capture thermal information from objects. They are used in applications such as night vision, temperature measurement, and heat detection.

1.7 USE OF COMPUTER VISION IN THERMOGRAPHY

Infrared (IR) based vision systems play a significant role in enhancing the field of Non-Destructive Testing (NDT) by providing unique capabilities for detecting defects, assessing material integrity, and ensuring the quality of components and structures.

Here's a detailed explanation of how IR-based vision systems can provide improvements in the field of NDT:

- Thermal imaging cameras, can detect defects that are not visible to the naked eye or conventional inspection methods. Thermal imaging captures temperature variations on the surface of objects, allowing inspectors to identify anomalies indicative of internal defects such as cracks, voids, delamination's, or inclusions. These defects often manifest as localized changes in temperature due to variations in material properties, internal stresses, or discontinuities within the material.
- IR-based vision systems provide quantitative data on thermal properties such as temperature, thermal conductivity, emissivity, and heat flow within materials. This information enables us to assess material integrity, characterize material properties, and identify deviations from expected thermal behavior that may indicate defects or abnormalities. By correlating thermal data with material properties, we can gain insights into the structural integrity, performance, and durability of materials in NDT applications.
- IR-based vision systems facilitate remote inspection of objects in hazardous or inaccessible environments, where direct contact or close proximity may pose risks to inspectors. Thermal cameras can capture images from a distance without the need for physical access, allowing inspectors to assess the condition of components in challenging or dangerous conditions such as high temperatures, confined spaces, or areas with toxic substances. Remote inspection enhances safety, minimizes exposure to hazards, and enables thorough assessment of assets without putting personnel at risk.
- IR-based vision systems can be integrated into automated inspection systems for continuous monitoring, real-time analysis, and rapid decision-making in NDT applications. By combining thermal imaging technology with advanced image processing algorithms and machine learning techniques, automated inspection systems can detect defects, classify anomalies, and trigger alerts or corrective actions as needed. Integration with automated systems improves inspection efficiency, reduces labor costs, and ensures consistent and reliable inspection results across large-scale industrial operations.

1.8 INTRODUCTION TO DEEP LEARNING

Deep learning, a subset of AI and ML, uses deep neural networks to automatically discover intricate patterns in vast datasets. Inspired by the human brain's structure, they have interconnected layers, and are known for its depth, which enables hierarchical feature learning. Various deep architectures, like CNNs and RNNs, excel in image analysis and sequence tasks. Deep learning excels at feature and representation learning and thus help it grow its knowledge base and apply them.

Specialized frameworks, like TensorFlow and PyTorch, streamline the development and deployment of deep models. Deep learning models are categorized based on architecture as, Feedforward Neural Networks (Simple structures for tasks like image classification and regression.), Convolutional Neural Networks (Ideal for image and video processing, using convolutional layers to learn features.) Recurrent Neural Networks (Designed for sequential data to capture time dependencies in tasks like NLP and speech recognition.), Gated Recurrent Unit (Efficient for sequential data with fewer gates but effective dependency capture.), Autoencoders (Used for unsupervised learning and data compression, helpful for tasks like image de noising.), Generative Adversarial Networks (Consist of a generator and discriminator for data generation, image synthesis, and style transfer). These architectural categories offer diverse models for different data and tasks offering a wide range to select the best fit for the intended application.

1.8.1 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a class of deep neural networks that have revolutionized the field of computer vision by enabling machines to learn and extract features directly from images or visual data. CNNs are specifically designed to handle tasks such as image recognition, object detection, segmentation, and classification. They have become the backbone of many state-of-the-art computer vision systems due to their ability to automatically learn hierarchical representations of visual data.

1.8.2 Key Components of Convolutional Neural Networks:

- **Convolutional Layers:**

The core building blocks of CNNs is convolutional layers. These layers consist of filters (also called kernels) that convolve or slide over the input image to perform feature extraction. Each filter extracts specific features from the input image by performing element-wise multiplication and aggregation operations. The output of each filter is called a feature map, which represents the presence of certain features or patterns at different spatial locations in the input image.

- **Pooling Layers:**

Pooling layers are used to down sample the feature maps obtained from convolutional layers while retaining important information. Common pooling operations include max pooling and average pooling, which reduce the spatial dimensions of the feature maps by taking the maximum or average value within small regions (e.g., 2x2 or 3x3) of the feature maps.

- **Activation Functions:**

Activation functions introduce non-linearity into the network, allowing CNNs to learn complex relationships and patterns in the data. Common activation functions used in CNNs include ReLU (Rectified Linear Unit), Sigmoid, and Tanh. ReLU is the most widely used activation function due to its simplicity and effectiveness in alleviating the vanishing gradient problem.

- **Fully Connected Layers:**

Fully connected layers, also known as dense layers, are used to perform classification or regression tasks based on the features extracted by convolutional and pooling layers. These layers connect every neuron in one layer to every neuron in the next layer, allowing the network to learn complex mappings between the extracted features and the output labels.

CNN architectures like LeNet-5, AlexNet, VGGNet, ResNet, InceptionNet, and MobileNet vary in depth, width, and connectivity patterns. Training a CNN involves initialization, forward propagation, loss computation, backpropagation, parameter optimization, and validation/testing. CNNs find applications in image classification, object detection, semantic and instance segmentation, face recognition, medical imaging, autonomous driving, and natural language processing. They offer powerful capabilities for feature extraction and representation learning, driving innovation across industries.

1.9 IMAGE SEGMENTATION

Image segmentation is a fundamental task in computer vision that involves partitioning an image into multiple regions or segments based on certain criteria, such as pixel intensity, color, texture, or spatial proximity. The goal of image segmentation is to group together pixels that belong to the same object or region while separating them from pixels belonging to other objects or backgrounds. Image segmentation plays a crucial role in various computer vision applications, including object detection, recognition, tracking, scene understanding, and medical image analysis.

1.9.1 Applications of Image Segmentation:

Image segmentation finds its use in various fields and proves to be efficient in retrieving the relevant data from images. Applications of Image segmentation in various fields include:

- **Medical Image Analysis:** Image segmentation is used in medical imaging applications for tasks such as tumor detection, organ segmentation, and tissue classification in MRI, CT, and microscopy images.
- **Autonomous Driving:** Segmentation is essential for understanding the environment in autonomous driving systems, enabling tasks such as road detection, lane marking detection, and object recognition.
- **Satellite Imagery Analysis:** Image segmentation is used in remote sensing applications for land cover classification, vegetation mapping, and urban area detection from satellite imagery.

- **Object Detection and Recognition:** Image segmentation is an integral part of object detection and recognition systems, allowing for precise localization and classification of objects within images or videos.
- **Biomedical Imaging:** Image segmentation is used in biomedical imaging for tasks such as cell counting, morphological analysis, and disease diagnosis in microscopy images.
- **Video Surveillance:** Segmentation is used in video surveillance systems for tracking and analyzing objects or people in crowded scenes, enabling tasks such as crowd monitoring and abnormal behavior detection. Thus, image segmentation is a fundamental task in computer vision that involves partitioning an image into meaningful regions or segments. It plays a crucial role in various applications across domains such as medical imaging, autonomous driving, remote sensing, and object recognition, enabling machines to analyze and understand visual data more effectively.

1.9.2 Various Image Segmentation techniques

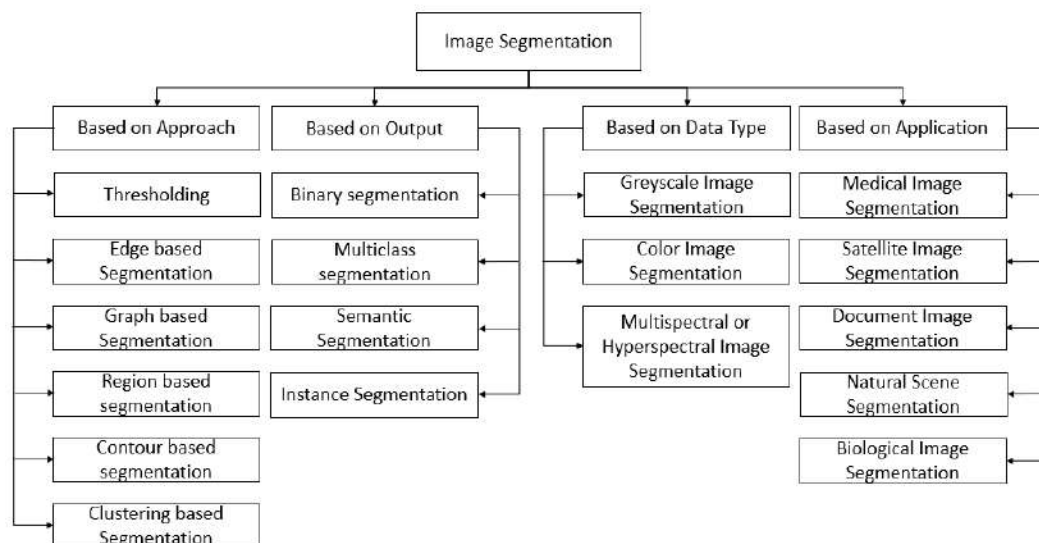


Figure 1.10: Image Segmentation techniques

A detailed classification of Image Segmentation techniques is done in Figure 1.10. Some of the main Image segmentation techniques include. Semantic segmentation is a specific type of image segmentation that assigns semantic labels to each pixel in an image, effectively dividing the image into multiple semantic categories.

Instance Segmentation: Instance segmentation is similar to semantic segmentation but

goes a step further by not only assigning class labels to pixels but also distinguishing between different instances of the same class. In instance segmentation, each pixel is assigned a class label as well as an instance ID, which indicates which specific object instance the pixel belongs to. Instance segmentation is commonly used in tasks such as object detection, where the goal is to precisely delineate and classify individual objects within an image.

Panoptic Segmentation: Panoptic segmentation combines both semantic segmentation and instance segmentation into a unified framework. In panoptic segmentation, each pixel is assigned a semantic label. The goal of panoptic segmentation is to provide a comprehensive understanding of visual scenes by simultaneously segmenting objects and regions based on both semantic and instance-level information.

Semi-Supervised Segmentation: Semi-supervised segmentation techniques leverage both labeled and unlabeled data to improve segmentation performance. These techniques typically involve training a segmentation model on a small set of labeled images and then using the trained model to generate pseudo-labels for unlabeled data. Semi-supervised segmentation can help overcome limitations of limited labeled data availability and improve the generalization ability of segmentation models.

Weakly Supervised Segmentation: Weakly supervised segmentation techniques train segmentation models using only coarse-level supervision, such as image-level labels or bounding boxes. These techniques aim to learn to segment objects or regions of interest without requiring pixel-level annotations. Weakly supervised segmentation methods often involve techniques such as attention mechanisms, saliency detection, or self-training to generate pixel-wise segmentations from weak supervision signals.

Interactive Segmentation: Interactive segmentation techniques involve user interaction to guide the segmentation process. These techniques typically provide users with tools or interfaces to manually annotate regions of interest or correct segmentation errors. Interactive segmentation can be useful in scenarios where automated segmentation methods may produce suboptimal results or require human oversight.

Hierarchical Segmentation: Hierarchical segmentation techniques partition an image into regions at multiple levels of granularity, forming a hierarchical segmentation tree or graph. These techniques aim to capture the hierarchical structure of objects and

regions in the image, from fine details to coarse structures. Hierarchical segmentation can facilitate multi-scale analysis and understanding of visual scenes, enabling tasks such as object recognition, scene parsing, and image synthesis. While semantic segmentation remains a prominent and widely used technique in computer vision, these related segmentation techniques offer alternative approaches and capabilities for addressing different segmentation challenges and application requirements. Depending on the specific task and context, researchers and practitioners may choose to explore and combine different segmentation techniques to achieve the desired segmentation performance and accuracy.

1.9.3 Advantages of Semantic Segmentation

Semantic segmentation offers several advantages over other types of image segmentation techniques, primarily due to its ability to provide fine-grained pixel-level understanding of visual scenes. Here are some of the key advantages of semantic segmentation:

- **Precise Object Localization:**

Semantic segmentation assigns a class label to each pixel in an image, allowing for precise localization of objects and regions of interest. Unlike bounding box-based approaches used in object detection, semantic segmentation delineates object boundaries at the pixel level, providing more accurate spatial information about object shapes and extents.

- **Detailed Scene Understanding:**

By segmenting an image into semantically meaningful regions, semantic segmentation facilitates detailed scene understanding and analysis. It enables the recognition and labeling of various objects, parts of objects, and background regions, leading to a comprehensive understanding of the visual content.

- **Pixel-Level Classification:**

Semantic segmentation classifies every pixel in an image into predefined semantic categories or classes, providing a rich representation of the scene at the pixel level. This pixel-level classification enables more nuanced and fine-grained analysis of visual

data, allowing for tasks such as object recognition, scene parsing, and image synthesis.

- Instance-Level Segmentation:

While semantic segmentation assigns class labels to pixels, it can also be extended to perform instance-level segmentation by distinguishing between different instances of the same class. Instance segmentation provides not only semantic information about objects but also delineates individual object instances, enabling tasks such as counting objects and tracking their movements.

- Semantic Contextual Information:

Semantic segmentation captures the semantic context of objects and regions within the image, considering the relationships and interactions between different classes. This contextual information can be valuable for higher-level reasoning tasks, such as scene understanding, contextual reasoning, and decision-making in autonomous systems.

- Input for Higher-Level Tasks:

Semantic segmentation serves as a fundamental building block for various higher-level computer vision tasks, including object detection, scene understanding, image captioning, and autonomous navigation. The accurate segmentation of objects and regions provided by semantic segmentation can significantly improve the performance of these downstream tasks.

- Applicability to Diverse Domains:

Semantic segmentation has broad applicability across diverse domains and applications, including medical imaging, autonomous driving, satellite imagery analysis, robotics, and augmented reality. It can be adapted and tailored to suit specific requirements and challenges in different domains, making it a versatile and widely used technique in computer vision.

Recent advancements in deep learning, particularly Convolutional Neural Networks (CNNs), have significantly advanced the SOTA in semantic segmentation. Deep learning-based approaches have achieved remarkable performance improvements in terms of segmentation accuracy, efficiency, and scalability, making semantic

segmentation more accessible and effective for real-world applications.

Overall, semantic segmentation offers a powerful framework for understanding and analyzing visual data at the pixel level, enabling a wide range of applications across various domains. Its ability to provide detailed scene understanding, precise object localization, and semantic contextual information makes it an indispensable tool in computer vision research and practical applications.

1.10 OBJECT DETECTION SYSTEMS

Object detection systems are computer vision algorithms designed to identify and locate objects within images or video frames. The primary goal of these systems is to recognize the presence of objects and accurately localize them by drawing bounding boxes around them. Object detection is a fundamental task in computer vision and plays a crucial role in various applications, including surveillance, autonomous vehicles, robotics, medical imaging, augmented reality, and more.

There are various components involved in the object detection systems.

- Localization helps in determining the precise spatial location of objects within an image or frame. This involves identifying the coordinates of the bounding boxes that enclose the objects.
- Classification involves assigning a class label to each detected object, indicating the category or type of object it belongs to. Common object categories include people, cars, animals, household items, etc. Object detection systems typically consist of several components that work together to detect and localize objects. These components may include Preprocessing steps such as image resizing, normalization, and enhancement may be applied to the input images or frames to improve detection performance.
- Feature extraction involves extracting relevant visual features from the input images or frames. This may include traditional handcrafted features or learned features extracted using deep learning techniques.
- Object proposal generation methods generate candidate regions or bounding boxes that may contain objects of interest. These regions serve as input to the detection algorithm, reducing the search space and improving efficiency.

The object detection algorithm analyzes the candidate regions and predicts the presence, location, and class of objects within each region. This may involve using machine learning models such as Support Vector Machines (SVMs), decision trees, or deep neural networks. Post-processing steps such as non-maximum suppression (NMS) may be applied to refine the detected bounding boxes, remove duplicate detections, and improve localization accuracy.

1.10.1 Requirements for Object Detection systems

Object detection systems have specific requirements to effectively identify and locate objects within images or video frames. These requirements ensure accurate detection and localization of objects, which is essential for various applications, including detecting weld defects in thermal images.

- **High Detection Accuracy:** Object detection systems should achieve high accuracy in detecting and localizing objects within images. This requires robust algorithms and models capable of accurately recognizing objects under various conditions, such as different lighting conditions, scales, orientations, and occlusions.
- **Real-time Performance:** For applications requiring real-time processing, such as surveillance, autonomous driving, or industrial inspection, object detection systems must operate efficiently and provide timely results. This necessitates the development of fast and computationally efficient algorithms capable of processing images or video frames in real-time.
- **Robustness to Variability:** Object detection systems should be robust to variations in object appearance, background clutter, noise, and other factors that may affect detection performance. This requires the ability to generalize well across different scenes, environments, and object categories.
- **Scalability:** Object detection systems should be scalable to handle large datasets, high-resolution images, and streaming video feeds. Scalability ensures that the system can effectively process and analyze large volumes of data without compromising performance or efficiency.
- **Adaptability:** Object detection systems should be adaptable to different applications

scenarios and requirements. This includes the ability to adapt to changes in object categories, scene conditions, and operational constraints through fine-tuning, transferlearning, or other techniques.

- **Interpretability:** Object detection systems should provide interpretable outputs that enable users to understand and interpret the results. This includes visualizing detected objects, their bounding boxes, and associated confidence scores, as well as providing explanations or justifications for detection decisions.

1.10.2 Applications in detecting Weld Defects in Thermal Images:

- **Identification of Defective Regions:** Object detection systems can be trained to recognize and localize weld defects within thermal images. By analyzing temperature variations or anomalies in the thermal data, these systems can identify regions with potential defects, such as cracks, porosity, lack of fusion, or excessive heat.
- **Early Detection and Prevention:** Detecting weld defects in thermal images allows for early identification of issues during the welding process or post-weld inspection. Early detection enables timely intervention and corrective actions to prevent defects from propagating or causing structural failures, leading to improved weld quality and reliability.
- **Automated Inspection:** Object detection systems can automate the inspection process by analyzing thermal images captured during welding operations or post-weld inspections. This reduces the reliance on manual inspection methods, improves inspection efficiency, and minimizes the risk of human error.
- **Improved Accuracy and Consistency:** Object detection systems offer higher accuracy and consistency compared to manual inspection methods, as they can analyze large volumes of thermal data rapidly and consistently. This reduces the likelihood of missed defects or false alarms, leading to more reliable inspection results.
- **Integration with Welding Systems:** Object detection systems can be integrated with welding systems or robotic welding platforms to provide real-time feedback and

control. By continuously monitoring thermal images during welding, these systems can adjust welding parameters, such as heat input or travel speed, to mitigate defects and optimize weld quality.

1.10.3 Types of Object Detection

Object detection algorithms can be categorized into different types based on their underlying methodologies, architectures, and techniques. Here are some common types of object detection algorithms:

- **Feature-Based Methods:**

Feature-based methods extract handcrafted features from regions of interest within the image and use machine learning algorithms to classify these regions as containing objects or background. Commonly used handcrafted features include Histogram of Oriented Gradients (HOG), Haar-like features, Local Binary Patterns (LBP), and color histograms. Feature-based methods often involve techniques such as Support Vector Machines (SVMs), decision trees, or other classifiers for object detection.

- **Deep Learning-Based Methods:**

Deep learning-based methods have revolutionized object detection by learning feature representations directly from data using deep neural networks. These methods leverage convolutional neural networks (CNNs) to extract hierarchical features from images and use them to predict bounding boxes and class probabilities. Deep learning-based object detection algorithms can be categorized into two main types: two-stage detectors and one-stage detectors. Two-stage detectors first generate a set of candidate object regions or proposals and then refine these proposals to produce final detections. One-stage detectors directly predict bounding boxes and class probabilities from fixed spatial locations within the image.

- **Hybrid Methods:**

Hybrid methods combine traditional feature-based techniques with deep learning approaches to leverage the strengths of both approaches. These methods may use handcrafted features for object proposal generation or initial feature extraction and then

refine the detections using deep neural networks. Hybrid methods aim to improve detection accuracy, efficiency, and robustness by incorporating complementary information from different sources.

- **Contextual and Graph-Based Methods:**

Contextual and graph-based methods incorporate contextual information and dependencies between objects to improve detection performance. These methods model the relationships between objects as graphs or graphical models and use techniques such as Conditional Random Fields (CRFs) to jointly infer object locations and classes based on contextual cues. Each type of object detection algorithm has its strengths and weaknesses, and the choice of algorithm depends on factors such as the complexity of the task, available computational resources, and specific requirements of the application. In practice, researchers and practitioners often experiment with different algorithms and architectures to achieve the desired balance of accuracy, efficiency, and scalability for their particular use case.

1.10.4 One step And Two step networks

In this era of deep learning, object detection is divided into two categories: "two-step network" and "one-step network".

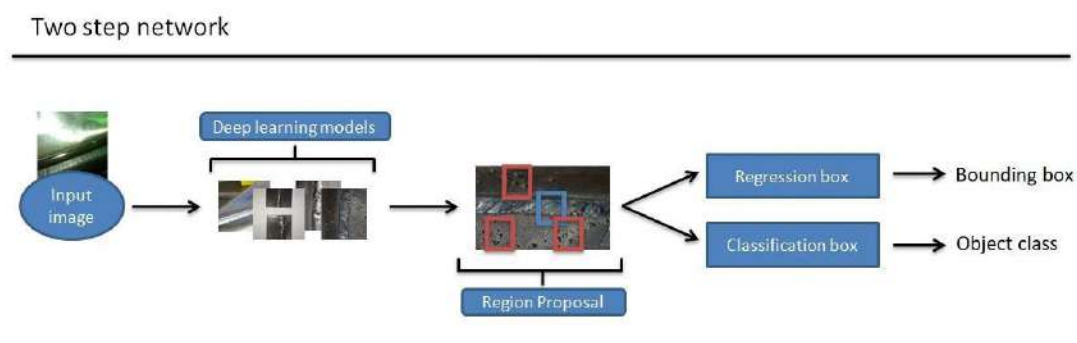


Figure 1.11: Two step networks

There are two approaches used in computer vision for detecting and localizing objects within images or video frames: one-stage and two-stage object detection. The two-stage approach, as seen in Figure 1.11, divides the detection process into two distinct stages. In the first stage, a region proposal network (RPN) or a similar method is used

to generate a set of region proposals. These region proposals represent areas in the image that are likely to contain objects. In the second stage, each region proposal is individually classified and refined to accurately localize the objects within those proposals. This approach, as demonstrated by Faster R-CNN, achieves high accuracy by carefully refining and classifying the region proposals independently.

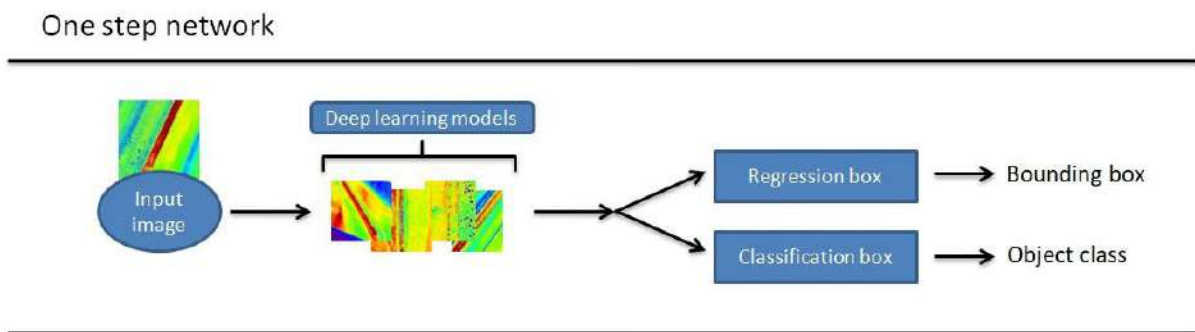


Figure 1.12: One step network

On the other hand, the one-step network as seen in Figure 1.12 combines the region proposal and classification/localization tasks into a single process, eliminating the need for a separate proposal generation stage. Models like YOLO (You Only Look Once) and SSD (Single Shot MultiBox Detector) exemplify this approach, as they directly predict bounding boxes and class probabilities for all objects in one go. This research is based on One-step network (Yolo V8). One-step network usually have faster inference times compared to two-step network, making them more suitable for real-time applications.

1.10.5 Object Detection algorithms

Object detection algorithms have evolved significantly over the years, driven by advancements in computer vision, machine learning, and deep learning. A brief overview of its evolution is given in Figure 1.13

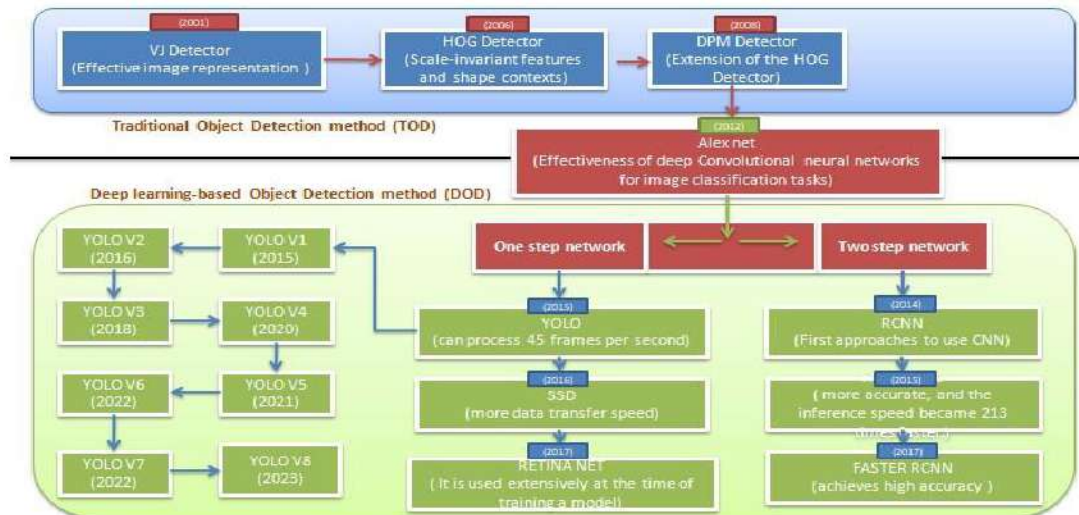


Figure 1.13: Evolution of Object Detection algorithms

- **YOLO (You Only Look Once)**

YOLO is a popular one-stage object detection algorithm known for its simplicity and real-time performance. YOLO divides the input image into a grid of cells and predicts bounding boxes and class probabilities for each grid cell. Each bounding box prediction includes coordinates (x, y) for the box center, width (w), height (h), confidence score indicating the likelihood of containing an object, and class probabilities. YOLO uses a single convolutional neural network (CNN) to simultaneously predict bounding boxes and class probabilities across the entire image in a single forward pass.

- **Faster R-CNN (Region-based Convolutional Neural Network):**

Faster R-CNN is a two-stage object detection algorithm that introduced the Region Proposal Network (RPN) to generate object proposals. The RPN generates a set of candidate object regions or proposals by sliding a small network (typically a few layers of convolutional operations) over the feature maps extracted from the input image. These proposals are then refined and classified by a separate detection network, which predicts bounding box coordinates and class probabilities for each proposal. Faster R-CNN achieves high accuracy by jointly optimizing the RPN and detection network within a unified framework.

- **SSD (Single Shot MultiBox Detector):**

SSD is a one-stage object detection algorithm that predicts bounding boxes and class probabilities directly from feature maps at multiple scales. SSD divides the input image

into a grid of default anchor boxes (or priors) at different aspect ratios and scales, and predicts offsets and confidence scores for each anchor box. The SSD architecture consists of a base network (such as VGG or ResNet) for feature extraction, followed by a series of convolutional layers for predicting bounding boxes and class probabilities. SSD achieves real-time performance by performing object detection in a single pass through the network and predicting detections at multiple scales simultaneously.

- Mask R-CNN:

Mask R-CNN extends Faster R-CNN by adding a branch for predicting segmentation masks in addition to bounding boxes and class probabilities. The mask branch operates in parallel with the bounding box branch and predicts pixel-wise segmentation masks for each detected object. Mask R-CNN shares the same backbone network and region proposal mechanism as Faster R-CNN but adds a mask prediction branch on top of the detection network. Mask R-CNN enables instance segmentation by providing accurate pixel-level segmentation masks for each detected object, in addition to bounding box coordinates and class probabilities.

- RetinaNet:

RetinaNet is a one-stage object detection algorithm designed to address the imbalance between foreground and background classes in training data, which often leads to poor performance in one-stage detectors. RetinaNet introduces a focal loss function that down-weights easy examples (i.e., well-classified negatives) during training, focusing more on hard examples that are misclassified with high confidence. The focal loss dynamically adjusts the loss weights based on the classification confidence scores, giving more emphasis to difficult-to-classify examples and improving the model's ability to handle class imbalance.

The key features made by each successive YOLO models is listed out and the improvements made in each model are listed out in Table 1.1 along with their respective Framework

Table 1.1: Key features of the YOLO models

Model	Framework	Key Features
V1 (2015)	DARKNET	Single-stage object detector. Combines object detection and classification into a single neural network architecture.
V2 (2016)	DARKNET	19 layers of CNN resulting in enhanced performance. Multi-scale training approach, prioritizing improved mean Average Precision (mAP) and high recall rates.
V3 (2018)	DARKNET	Integrates YOLO-v2, Darknet-53, and residual networks, Incorporates skip connections within its architecture.
V4 (2020)	DARKNET	Improved feature aggregation methods to extract significant features for precise object detection while aiming to retain a lightweight design.
V5 (2020)	PYTORCH	Built on the PyTorch framework, the architecture becomes more accessible. Facilitates deployment and testing across multiple platforms.
V6 (2022)	PYTORCH	Anchor-free approach, 51% increase in speed compared to anchor-based methods. Introduces updated reparametrized neck (Rep- PAN) modules and backbone (EfficientRep), enhancing feature extraction and aggregation.
V7 (2022)	PYTORCH	Integrates the E-ELAN (extended efficient layer aggregation network) into its backbone architecture. Introduces the Trainable BoF for enhanced detection performance during training.
V8 (2023)	PYTORCH	In comparison to its predecessors has demonstrated improved performance in terms of throughput when evaluated with similar computational parameters highlighting the importance of fast detection, high accuracy, and deployment on constrained edge devices.

CHAPTER 2

LITERATURE SURVEY

2.1 LITERATURE SURVEY BASED ON THERMAL IMAGING SYSTEMS

Based on the relevant research work on thermography-based defect detecting applications, a detailed categorization of Material, Camera, implemented deep learning model has been made. Various combinations have been made by using thermal camera and deep learning models which provide various benefits and provide accuracies, errors and speed respective to them.

Table 2.1: Categorization based on Material Application, Camera used, Implemented models

Reference No.	Authors & year	Material application	Camera used	Implemented models or algorithms	Inference
[22]	Wael M. Elmessery	Broiler Pathological Phenomena	UTi165A thermal camera	Convolution neural network (CNN). Yolo-based	- The YOLO V8-based model offers efficient. - This study's findings pave the way for advancements in precision livestock farming and highlight the potential of AI-based detection systems to improve poultry production and animal care.
[1]	Ali Sarhadi, Rodrigo Q. Et al (2021)	glass-epoxy composites	HTI HT-18 camera	Kemel Ridge Regression (KRR)	- Through cross-validation, accurately predicted point source data with low errors. - However, the model couldn't locate defects precisely and assumed specific shapes.
[2]	Angela Busheska, Et al (2022)	pathologies in buildings	HTI HT-18 & FLIR One Pro cameras.	Convolutional Neural Network (DCNN)	- Image processing with MatLab included noise reduction and quality enhancement. - DCNN. Results, including accuracy, precision, recall, and F1 score, were measured using 80% training and 20% validation sets. - The accuracy exceeded 96%, but recognizing first-level cracks posed challenges.
[7]	Hyun-Tae Bang, Et al (2020)	carbon/organo fabrics	FLIR A65 infrared camera	The Faster RCNN method (CNN)	- They use a Faster RCNN model for defect identification in three steps CNN image processing, RPN-based defect location proposals, and defect identification with confidence levels. - InceptionV2 is used for feature extraction. The dataset is augmented, and transfer learning is applied using the COCO dataset.
[11]	Roberto Marani, Et al (2021)	glass-fiber-reinforced polymer	Flir A 655sc uncooled IR camera	Convolutional Neural Network (CNN) Architecture	This approach integrates step-heating thermography with CNN based thermal signal processing for defect detection and classification.

[23]	Yunze He	Car, Electrical equipment, Composites, Rotating machine and etc.	LWIR FLIR T650SC and FLIR One	Convolutional Neural Network (CNN) Architecture	• IRMV includes passive and active thermography. • This paper covers IRMV principles, cameras, data, DL applications, and trends on various platforms, focusing on mobile-based DL models.
[15]	Vijay John, Et al (2021)	Object Detection in Autonomous Driving	FLIR Tau 2	RV-net, TV-net and Yolo. (CNN)	• These networks integrate vision-based data with radar or thermal camera inputs and feature two input branches for sensor-specific feature extraction, along with an output branch that employs the tiny YOLO network for classification.

The categorization in table 2.1 provides us detailed view on how the deep learning model used affects the accuracy, speed and error generated by the model. Based on the cameras used FLIR A65 provides better resolution of the thermal images and frame rate compared to others while HTI HT-18 and FLIR series cameras are cheaper alternatives to it. The deep learning models used here depend upon Convolutional Neural Networks and each have their own advantages. The deep learning models RCNN and DCNN have main key differences in their operation. The DCNN model training is time consuming and computationally intensive due to its complexity whereas in RCNN the original models were slower in both training and deducing conclusions. Though DCNN lack in fine grained localization they have high accuracy in classification of the image. RCNN models are slower than the DCNN models but they excel in accurately locating the objects in images. YOLO has combined both of these architectures to give a new model YOLO V8 which has high accuracy and localization of objects in comparison to these deep learning models.

2.2 LITERATURE SURVEY BASED ON YOLO MODELS

YOLO model due to its rein as SOTA model is preferred for its speed, efficiency, and accuracy in object detection. Unlike other models, it detects objects in a single pass, making it suitable for real-time applications. Its unified framework simplifies implementation and deployment compared to more complex models. The following research conducted based on the use of YOLO and comparing it to other models helps us to understand the working of YOLO deep learning models and how its beneficial.

Ángel Morera (2020) conducted a comparative study of the SSD and YOLOv3 deep neural network models for detecting outdoor urban advertising panels in images. The research conducted shows that while SSD excels in minimizing False Positive cases,

YOLOv3 demonstrates superior panel localization by detecting a higher number of True Positive panels with increased accuracy. YOLOv3 outperforms SSD in detecting small panels and in challenging scenarios like partial occlusion, rotation, and night illumination. The study also includes a comparison with semantic segmentation networks, highlighting the strengths and weaknesses of each model under different conditions. Overall, YOLOv3 proves to be a promising choice for accurate and efficient detection of outdoor advertising panels in urban environments.

Tausif Diwan (2022) thoroughly examined single-stage object detectors, particularly focusing on the YOLO framework, covering regression formulation, architectural enhancements, performance metrics, and a comparison with two-stage object detectors. It analyses YOLO frameworks in detail, including incremental architectural improvements, optimization strategies, loss function significance, and real-world applications. The comparative performance of two-stage and single-stage object detectors, specifically different YOLO versions, is evaluated, along with exploration of four iterations of YOLO and enhancements in each version. The paper emphasizes the importance of single-stage object detectors, notably YOLO frameworks, in object detection, providing insights into their challenges, advancements, and practical applications. It serves as a valuable reference for researchers and practitioners in the field, offering a comprehensive analysis of single-stage object detectors and their contributions to detection technologies.

Matko Glucina (2023) though his research assessed various YOLOv5 models, including nano, small, medium, and large, for their effectiveness in detecting and segmenting PCBs. Results revealed that the medium model (YOLOv5m) achieved high precision, recall, and mean average precision (mAP) values for both bounding box and segmented mask detection. The study discussed how the choice of backbone model size influences results and suggested selecting models based on object size and complexity. The research uses an open-source dataset from Kaggle containing 1300 images which were annotated in roboflow software into 13 classes. The YOLO V5 models (s, m, l) were trained using these annotated images. Each model achieved results in the range of 0.9 for mean, precision, recall, Map and F1-score.

Martin Drahanský (2023) developed a thermal imaging system using a Lepton 5 camera and Raspberry Pi 3B to detect and locate individuals. It served various purposes like monitoring people flow in museums, managing smart homes, and overseeing hazardous areas such as railway platforms. Components include a control

and capture library for the Lepton 5 and YOLO V4 object detector for person detection. Testing at the Czech National Museum highlighted its versatility and effectiveness indoors and outdoors, with support from the v4l2lepton3 library. FLIR's dataset of over 14,000 thermal images, primarily featuring people, was used, with annotations in COCO format. The challenge lies in the camera's low resolution, where accuracy diminishes notably with distance. A one-pixel difference in the image could result in a one-meter variance in location estimation at approximately 16 meters from the camera. Even slight adjustments in the bounding box around an individual can lead to significant location deviations.

CHAPTER 3

AIM AND SCOPE

3.1 AIM

The aim of the research is to develop a Yolo based state of the art image segmentation model to identify weld defects on thermal images using a thermal imaging setup created by using Raspberry Pi 4 and Lepton 1R.

3.2 OBJECTIVE

- Identity and collect images of several types of defects in welded materials using the infrared imaging system.
- Annotate them and augment them to generate more samples
- Creating a dataset containing these images to train the YOLO V8 model.
- Create a deep learning-based algorithm using YOLO V8 and semantic segmentation to process thermal images generated by the active infrared imaging kit.
- Validate them against test dataset
- To detect the defects in metal components in real time using the infrared imaging kit integrated with the algorithm.

3.3 SCOPE OF THE INVESTIGATION

Detecting weld defects is crucial for ensuring the quality of welded components, as these defects can compromise structural integrity and lead to potential failures. By identifying defects, welds can be brought into alignment with industry standards and specifications, thereby enhancing the overall quality of welded joints. Additionally, the detection of defects plays a vital role in safety, particularly in critical sectors such as aerospace, automotive, and construction, where defective welds pose significant hazards. Early detection allows for timely repairs or adjustments, reducing the need for costly rework or component replacement and resulting in savings in materials, labor, and downtime. Moreover, compliance with regulatory standards governing weld quality

and safety is essential across various industries to avoid potential legal and financial consequences. Diligent defect detection is essential for maintaining safety, quality, compliance and cost-effectiveness. Thermal imaging systems proves to be helpful by detecting weld defects by capturing thermal signatures, revealing anomalies like cracks and porosity. They use infrared technology to visualize temperature variations, enabling quick and accurate assessment of weld quality without disrupting component integrity. These systems offer real-time monitoring during welding, allowing prompt corrective action and enhancing overall quality assurance and safety in welding operations. YOLO being a deep learning algorithm that can detect objects in images or videos with high accuracy and speed. In the context of welding defect detection, YOLO can analyze thermal images captured by cameras in real-time and accurately identify defects such as cracks, porosity, lack of fusion, and inclusions. Its ability to process images quickly makes it suitable for monitoring the welding process continuously, allowing for timely intervention to address defects as they occur. Additionally, YOLO's versatility and ease of implementation make it a practical solution for integrating defect detection into existing welding systems or workflows.

CHAPTER 4

EXPERIMENTAL SETUP AND METHODS

4.1. OVERALL PROCESS FLOW OF THERMAL IMAGING BASED DEFECT DETECTION SYSTEM

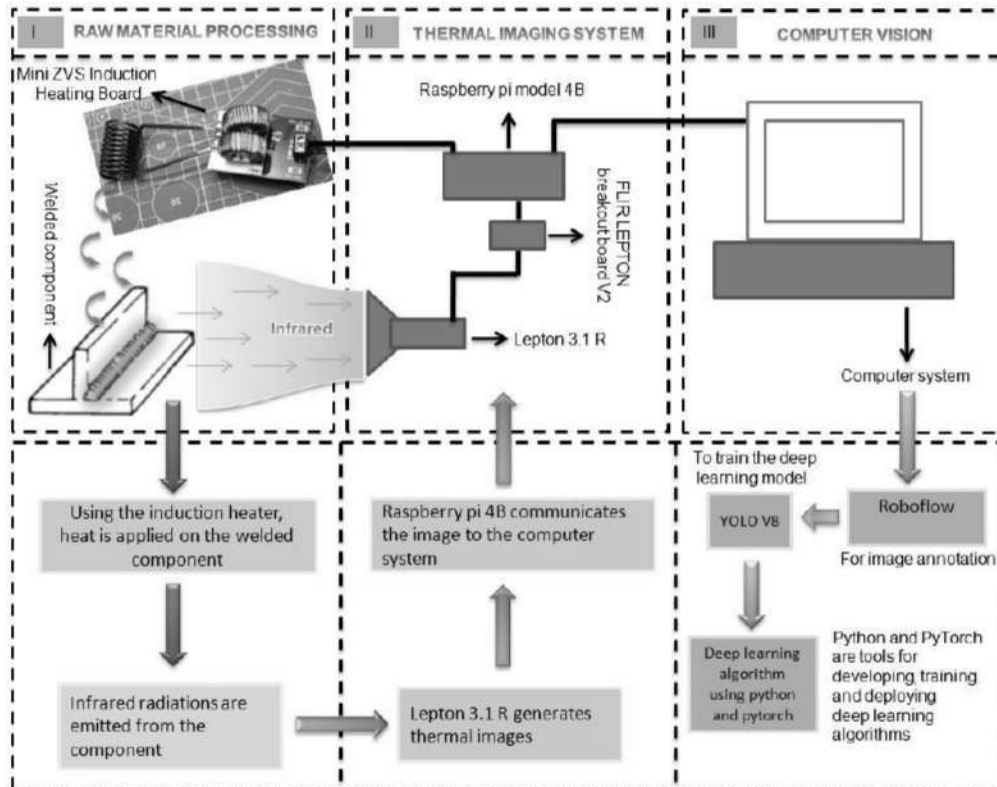


Figure 4.1: Proposed Defect Detection System

The research focuses on creating a YOLO v8 based semantic segmentation model as plotted out in Figure 4.1 to identify weld defects. YOLO v8 being the latest state-of-the-art YOLO model, when compared with YOLO-v5 and YOLO-v6 exhibits improved performance suggesting advancements in hardware-efficient architecture. YOLO also known as you only look once reigns itself as the state-of-the-art model and has been proven to be the best among the object detection models. YOLO v8 has been built based upon the previous versions of it. It uses Convolutional Neural Network (CNN), consisting of two parts the head and the backbone. The backbone is a modified version of CSPDarknet5 It consists of 53 convolutional layers enabling information flow through cross stage partial connections. The head of YOLOv8 is composed of several convolutional layers along with a sequence of connected layers. These components are tasked with predicting bounding boxes, objectness scores, and class probabilities

for identified objects within the image. The images required for this model is processed through Roboflow which enables us to annotate the images usefully and efficiently. Roboflow is a platform that simplifies the process of building and managing images for deep learning models. It allows us to upload datasets, preprocess images, annotate them and generate training data in various formats. Roboflow also provides tools for evaluating model performance and deploying models to production environments. It streamlines the entire workflow of developing and deploying computer vision models.

4.2 EXPERIMENTAL SETUP FOR THERMAL IMAGING SYSTEM

The Raspberry Pi 4 Model B is a small single-board computer (SBC) developed by the Raspberry Pi Foundation. The Raspberry Pi 4 boasts a quad-core ARM Cortex-A72 processor, offering substantially improved performance compared to previous models. This makes it suitable for a broad spectrum of applications, including multimedia, gaming, and desktop computing. With dual micro-HDMI ports, the Raspberry Pi 4 can concurrently drive two displays with 4K resolution, making it well-suited for multimedia and digital signage applications. The inclusion of Gigabit Ethernet, dual-band Wi-Fi, and Bluetooth 0 ensures faster network connectivity and enhanced wireless capabilities compared to its predecessors. USB 0 ports enable faster data transfer rates, facilitating efficient interaction with external storage devices, peripherals, and accessories. The Raspberry Pi 4 incorporates improved thermal management mechanisms, resulting in more efficient heat dissipation. This ensures stable performance even under demanding workloads and diverse operating conditions.

Embedded Systems: Due to its compact size, affordability, and versatility, the Raspberry Pi 4 finds applications in embedded systems and prototyping projects across various industries, including aerospace, automotive, healthcare, and industrial automation. With its upgraded performance and support for dual 4K displays, the Raspberry Pi 4 serves as an economical desktop computer solution for performing basic productivity tasks. The key specifications of Raspberry Pi are listed out in table 4.1.

Table 4.1: Specification of Raspberry PI 4

Raspberry Pi 4	Specifications
CPU	Quad-core ARM Cortex-A72
RAM	2GB, 4GB, or 8GB LPDDR4
GPU	Video Core VI
USB Ports	4 x USB 0, 2 x USB 0
Video Output	2 × micro-HDMI ports
GPIO Pins	40 GPIO pins
Operating System	Raspberry Pi OS

4.3 WORKING OF THE THERMAL IMAGING SYSTEM

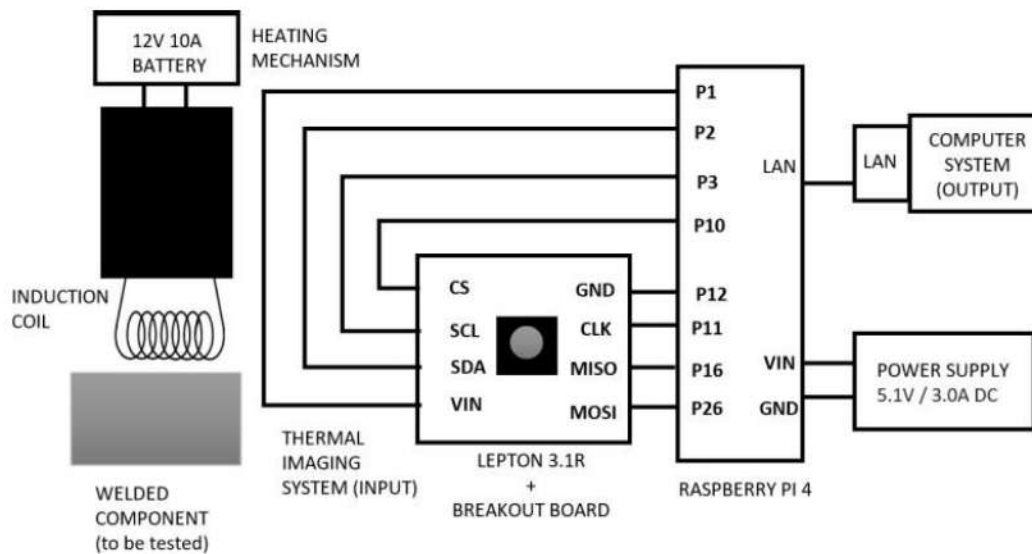


Figure 4.2: Circuit Connection Diagram of the thermal imaging setup

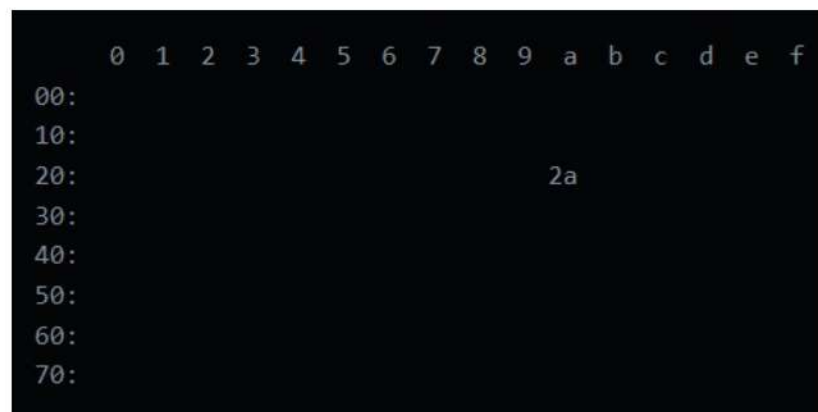
The Raspberry Pi 4 is connected to induction coil, battery and Lepton 3.1 R camera based on the Figure 4.2 using jumper cables and wires. The connections are verified and the Power supply is given to the Raspberry Pi. The Raspberry Pi needs a separate display but this issue can be overcome by enabling VNC in Raspberry Pi. VNC (Virtual Network Computing) on Raspberry Pi provides a means to remotely access and manage the Raspberry Pi's graphical desktop interface from other devices like computers, tablets, or smartphones. It enables us to interact with the Raspberry Pi, irrespective the connections wirelessly. To employ VNC on Raspberry Pi, we need to install and configure VNC server software directly on the Raspberry Pi. This software

enables the Raspberry Pi to share its desktop environment over a network connection. Once the VNC server setup is complete, a VNC viewer application on the remote device facilitates connection to the Raspberry Pi. Upon establishing the connection, the user can view the Raspberry Pi's desktop interface on their remote device's screen, interacting with it via the device's keyboard, mouse, or touchscreen. The VNC makes our thermal imaging system flexible to use.

Upon connecting the raspberry pi to our Laptop, we can interact with the Raspberry Pi to obtain images. The images are taken using the lepton packages available from the official site. The packages are installed and setup in the Raspberry Pi allowing us to take images using the Raspberry Pi.

There are 3 main lines of codes used to interact with the Raspberry Pi they are:

- The first code is “`i2cdetect -r -y 1 0x2a 0x2a`” this line enables us to identify if the camera is working or not. is used in the Raspberry Pi terminal to perform I2C (Inter-Integrated Circuit) bus scanning. It scans the I2C bus 1 (-y 1) for devices with addresses in the range of 0x2a to 0x2a (0x2a 0x2a). This command checks for any devices connected to the I2C bus at the specified address and displays their presence in the terminal output. If the camera is working the matrix as per Figure 4.3 will be generated. The 2a denotes that the lepton is able to respond to the Raspberry Pi.



	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00:																
10:																
20:											2a					
30:																
40:																
50:																
60:																
70:																

Figure 4.3: Matrix generated on the Raspberry Pi denoting that the Raspberry Pi responds.

- The second code is “`rpi_vsync_app` “. This code enables us to see if the Raspberry Pi is able to successfully send and receive signals with the Lepton 3.1R camera. It is a software application and library designed to

interface with the Lepton camera module and Raspberry Pi and perform various tasks related to thermal imaging, such as capturing and processing thermal images. It enables us to integrate the camera module into our proposed system. Running this code gives a bunch of lines the last 6 lines are read to see if the signals are read properly. The program runs for a few seconds. From Figure 4.4 we observe that the last line is “bc 45 “ and “LEP_GetOemGpioMode gpio_mode = 5 result = 0”. This shows that the Raspberry Pi can send and receive signals from the Lepton 3.1R camera.

```
i2cdev_read_byte_data(rx_adr=0x25e80, rx_data=0x25670, rx_size=2)
Writing big-endian address 0x0028
Read back 2 bytes from I2C read:
bc 45

LEP_GetOemGpioMode gpio_mode = 5 result = 0.
```

Figure 4.4: Lines generated by Raspberry Pi using rpi_vsync_app

- The Final code used in Raspberry Pi is `lepton_data_collector -3 -c 50 -o /tmp/capture/frame_` this code enables us to take thermal images in greyscale using the Lepton 3.1R camera. The obtained images are stored in the directory specified which is `tmp/capture/frame_`. The data collector runs for a few seconds and takes about 50 images and stores in the capture folder.

Once the process is finished, there will be a collection of 50 images captured from the Lepton 3.X camera module, equivalent to approximately 5 seconds of footage at around 9 frames per second (fps). These images will be named sequentially from `/tmp/capture/frame_000000.gray` to `/tmp/capture/frame_000049.gray`. To view these images on a Linux system, users can utilize the ImageJ Java application. After launching ImageJ, users should navigate to the File menu, select Import->Raw..., and configure the image type to 32-bit unsigned, specifying the appropriate width and height (160x120) in the ensuing dialog box. By checking the Open all files in folder checkbox, users can import the entire sequence into an image stack. This facilitates the creation of a .avi movie from the frames using the File->Save As->AVI... option, where they can set the frame rate to 9 fps for smooth playback as shown in Figure 4.5.

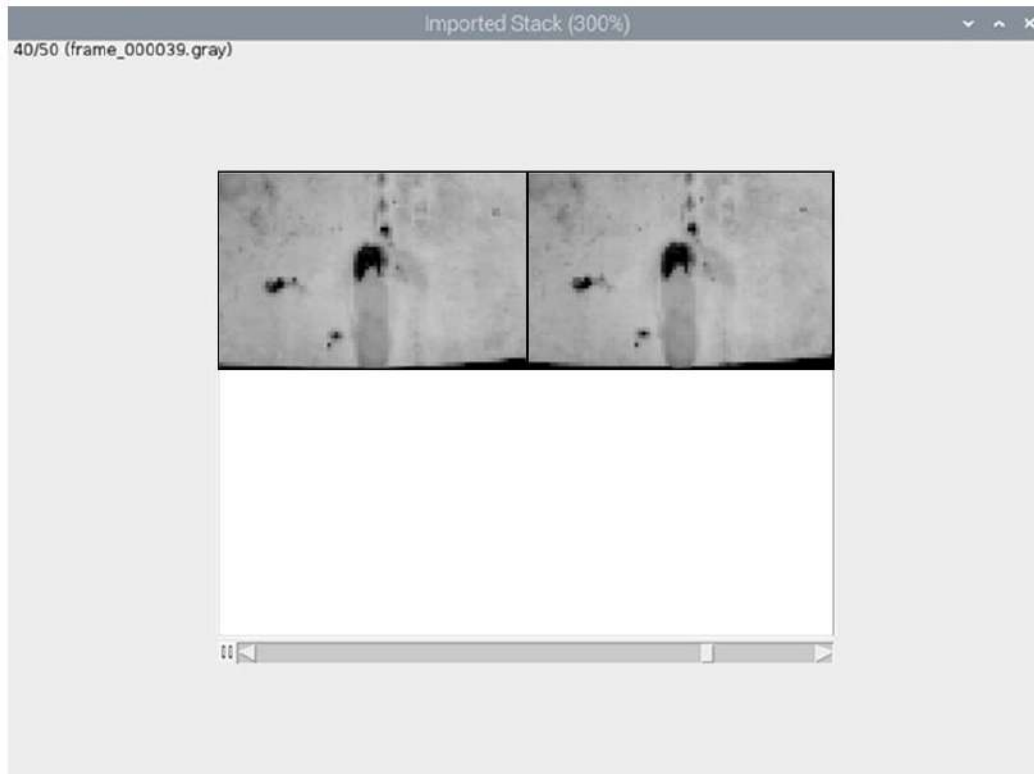


Figure 4.5: Thermal image obtained using the Thermal Imaging Setup

The images received from the lepton 1R are then used to generate the deep learning model using YOLO V. These generated images are communicated to the computer system using the Raspberry Pi. The deep learning model is created with the help of YOLO V8 which is a State-Of-The-Art (SOTA) model known for its speed, accuracy, and easy access, makes it an excellent choice for image segmentation. The generated images are preprocessed with the help of roboflow software to create the dataset for the deep learning model. Roboflow offers many features including Data Annotation, Augmentation and Versioning which assist us to annotate the images enhance its quality and keep track of the data in the dataset. Roboflow supports semantic segmentation annotation, in which every pixel in an image is associated with a specific label or tag. This annotation process is crucial for this research as the data set requires detailed pixel-level understanding. Thus, a high-quality training dataset is created for the deep learning model enabling accurate segmentation. The image data set is then used in YOLO V8 to generate the Deep Learning model. The created Deep Learning model is subsequently used in the python program to detect defects in real time utilizing the experimental setup.

4.4 DATA PREPARATION AND AUGMENTATION OF THERMAL IMAGES

The main requirement for creating a defect detection model includes the image dataset. The image dataset used for training the model was created using a preexisting dataset from Kaggle (open-source platform). The image dataset was used to create a new image segmentation set allowing us to use it for image segmentation in YOLO v8 model. The new set consists of an even distribution of images allowing us to produce the training analytics and compare them. The distribution of these images are as per table 4.2.

Table 4.2: Distribution of Images used for creating the dataset to train the deep learning model

Class	Train	Valid	Test
Porosity	359	33	41
Crack	80	20	10
Overlap	80	20	10
Slag Inclusion	80	20	10
Lack of fusion	80	20	10

The images are first preprocessed and then augmented in Roboflow. The Roboflow platform to manipulate and enhance images for various computer vision tasks. With Roboflow, we can perform a range of image augmentation techniques, such as flipping, rotating, resizing, adjusting brightness and contrast, adding noise, and applying geometric transformations like scaling or shearing.

To begin augmenting images with Roboflow, the dataset needs to be uploaded to the platform and then the desired augmentation techniques from the available options are selected. Once the augmentation parameters are set, Roboflow processes the images accordingly, generating augmented versions that can be used to improve the performance of machine learning models. The augmentations used for this set were flipping them horizontal and vertically. The images were also rotated clockwise, anticlockwise and upside down. Figure 4.6 mentions clearly the preprocessing and augmentation of the weld defect data set.

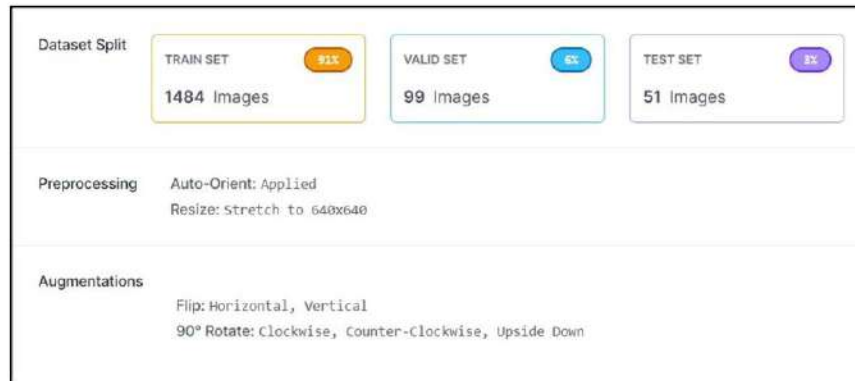


Figure 4.6: Images preprocessed and augmented in Roboflow

Augmenting images using Roboflow can help increase the diversity and robustness of the dataset, leading to better model generalization and performance. Additionally, it can help address issues like overfitting and data scarcity by generating additional training samples. Overall, Roboflow provides a convenient and efficient way to augment images for various computer vision applications, making it a valuable tool for data preprocessing and model training.

4.5 ANNOTATION OF IMAGES USING ROBOFLOW

The annotated images are then pre-processed to orient the images and resized to 640 x 640 pixels to make them uniform. The images were additionally flipped and rotated to provide variations resulting in a completed Yolo V8 model training dataset consisting of 1484 for training, 99 for validating and 51 images for testing. After selecting the dataset, the images are annotated using the "Annotate" tab to open the Roboflow Annotator tool. The images are presented one by one enabling us to annotate the images with a set of drawing tool. The provided drawing tools are utilized to annotate objects in the images, such as creating bounding boxes or employing other supported annotation types like polygons or key points. Following annotation of an image, save the annotations by clicking on the "Save" button. You can navigate between images using either the arrow buttons or the image thumbnails displayed on the side. Once all images in the dataset have been annotated, review them to confirm accuracy. Additionally, adjustments can be made to annotations by selecting and modifying them as necessary. These images were then annotated in ROBOFLOW marking these defects accurately. As per the Figure the images were annotated for a total of 5 defects.

They were for porosity, slag inclusion, Lack of Fusion, Crack and Overlap. These defects were carefully analyzed and annotated. The annotations with their specific class color is observed in Figure 4.7.

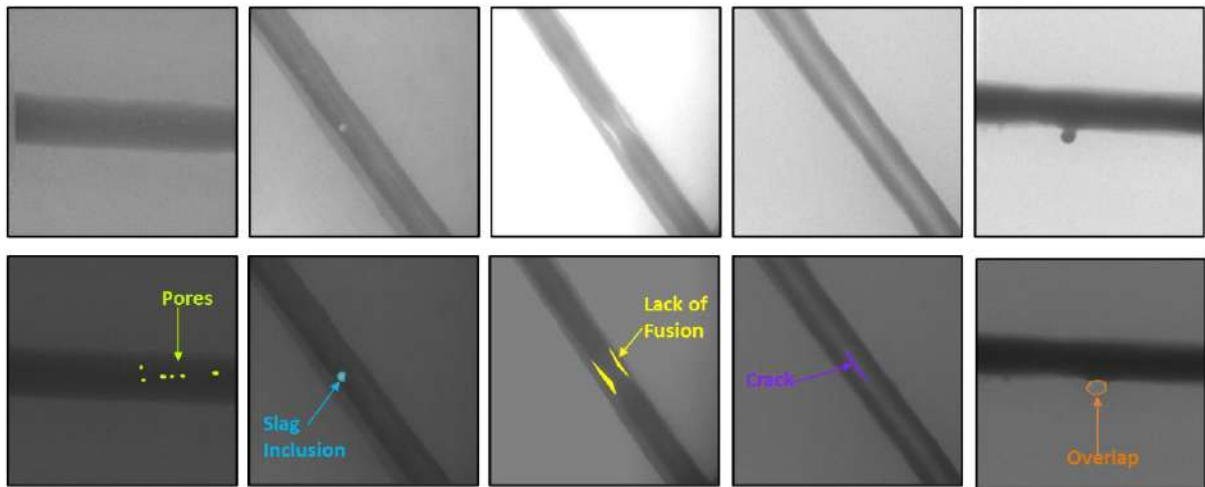


Figure 4.7: Image Annotated in Roboflow for their respective defects

Roboflow allows for seamless integration with YOLO v8 by allowing us to download the image dataset and also provides a coding snippet, as shown in Figure 4.8 to be used in roboflow to import the dataset. The code is then used in Google Collab to train the YOLO v8 model. The Object segmentation model was created using yolov8m-seg, the model for creating yolo v8 mediumsegmentation models. The Defect detection model was successfully trained and the results are discussed.

```
!pip install roboflow

from roboflow import Roboflow
rf = Roboflow(api_key="6mORbBW7ivlSUIDXxQsT")
project = rf.workspace("cassius-fro-9ykox").project("project-qpf6k")
version = project.version(2)
dataset = version.download("yolov8")
```

Figure 4.8: Code snippet from Roboflow used in Google Collab

4.6 YOLO (OBJECT DETECTION ALGORITHM)

YOLO, short for "You Only Look Once," is a popular object detection algorithm known for its simplicity, efficiency, and real-time performance. YOLO was introduced by

Joseph Redmon et al. in 2016 and has since undergone several iterations, with YOLOv3 being one of the most widely used versions. Here's a detailed explanation of how YOLO works, its architecture, and its applications: YOLO takes a different approach compared to traditional object detection methods by framing object detection as a single regression problem, rather than multiple classification and localization tasks. Instead of dividing the image into a grid of cells and predicting bounding boxes for each cell individually, YOLO directly predicts bounding boxes and class probabilities for a predefined set of anchor boxes across the entire image.

4.6.1 Key steps involved in the workflow of YOLO

- **Input Processing:** YOLO takes an input image of fixed size and divides it into a grid of cells. Each grid cell is responsible for detecting objects whose center falls within the cell.
- **Feature Extraction:** YOLO uses a deep convolutional neural network (CNN) to extract feature maps from the input image. The CNN architecture typically consists of multiple convolutional layers followed by max-pooling layers to down sample the spatial dimensions of the feature maps while increasing the number of channels.
- **Prediction:** YOLO predicts bounding boxes and class probabilities for each grid cell. For each anchor box associated with the grid cell, YOLO predicts, the coordinates (x, y) of the bounding box center relative to the grid cell, The width (w) and height (h) of the bounding box relative to the entire image, A confidence score indicating the likelihood that the bounding box contains an object, Class probabilities for each object class.

YOLO applies non-maximum suppression to remove redundant bounding box detections and retain only the most confident predictions. NMS suppresses bounding boxes with low confidence scores or high overlap with other bounding boxes.

4.6.2 Architecture of YOLO

The architecture of YOLO typically consists of several convolutional layers followed by

a final convolutional layer responsible for predicting bounding boxes and class probabilities. YOLOv3, for example, has a backbone network based on the DarkNet architecture, which includes 53 convolutional layers. The final layer of YOLOv3 predicts bounding boxes using three different scales (or anchor boxes) to handle objects of various sizes. **Backbone Network:** The backbone network extracts hierarchical features from the input image and generates feature maps with spatial information. **Detection Head:** The detection head consists of convolutional layers responsible for predicting bounding boxes and class probabilities for each anchor box associated with the grid cells. **Loss Function:** YOLO uses a combination of localization loss, confidence loss, and classification loss to train the model. The loss function penalizes deviations between the predicted bounding boxes and the ground truth annotations, as well as incorrect class predictions.

YOLO has found applications in various domains and use cases, including: **Object Detection:** YOLO is widely used for detecting objects in images and video streams in real-time. It is used in applications such as surveillance, autonomous driving, robotics, and security systems. **Vehicle Detection:** YOLO can be used to detect vehicles in traffic surveillance systems, parking lot monitoring, and vehicle counting applications. **Person Detection:** YOLO is effective at detecting people in crowded scenes, pedestrian tracking, and human activity recognition tasks. **Industrial Inspection:** YOLO can be applied to detect defects in manufacturing processes, such as detecting anomalies in product assembly lines or identifying faults in industrial equipment. **Medical Imaging:** YOLO can assist in medical image analysis tasks, such as detecting abnormalities in radiological images, segmenting organs, and identifying tumors in MRI or CT scans. Overall, YOLO offers a fast, efficient, and accurate solution for object detection tasks, making it suitable for a wide range of applications across different domains. Its real-time performance and simplicity make it particularly appealing for scenarios where fast and reliable object detection is required.

4.6.3 Training of Deep Learning Model using YOLO V8

A deep learning model is trained with the help of the created image dataset containing the thermal images of the weld defects which were clearly annotated using the ROBOFLOW software. The annotated images are used to train the model in the form of a code snippet generated from ROBOFLOW. The code snippet enables for

seamless deployment of the image dataset for any application.

The YOLO V8 model was trained in a google collab notebook allowing us to access remote GPU, enabling us to have better performance and speeds to train the model faster. The GOOGLE COLLAB notebook had the version python3 and Python 3 Google Compute Engine backend (GPU). It enabled us to have access to RAM of 15GB and a Disk Space of 78.2GB. The required packages for the model development are also downloaded like the Ultralytics library which contains the necessary files for using YOLO V8 and Roboflow library to read the images in the dataset. The setup of the environment is done and therefore the model can be trained in the GOOGLE COLLAB notebook.

The YOLO model offers input sizes categorized into small (S), medium (M), large (L), and occasionally extra-large (XL) or other variants, though the exact dimensions for each category can vary depending on the YOLO version and implementation. In the small (S) category, input sizes typically range around 320x320 or 416x416 pixels, ideal for situations with limited computational resources or requirements for real-time processing. Medium (M) input sizes, falling within the approximate range of 512x512 to 608x608 pixels, strike a balance between accuracy and computational efficiency, suitable for a wide array of applications. On the other hand, large (L) input sizes, such as 832x832 or 1024x1024 pixels, offer heightened accuracy but may necessitate more significant computational resources for inference. YOLO implementations may feature extra-large (XL) or other sizes like 1280x1280 or 1536x1536 pixels, tailored to specific scenarios where utmost precision is crucial, albeit at the cost of increased computational complexity. Choosing the appropriate input size depends on various factors, including the specific needs of the application, available computational capabilities, and desired trade-offs between processing speed and accuracy.

The YOLO V8 model used for training the defect detection model was `yolov8m-seg.pt` which is a medium sized model used for image segmentation. The number of epochs is also setup here. In machine learning and deep learning, an epoch is a complete pass through the training dataset. During each epoch, the model sees every example once and adjusts its parameters based on the error between predictions and true labels. The number of epochs set for this model is 20. The image size is also specified here to be 640px. Using these data the model is trained on the image dataset. Figure 4.9 shows the completion of the training of the model along with its specification and accuracies.

```

As speeds compiled in 0.174 hours.
Optimizer stripped from runs/segment/train/weights/last.pt, 54.0MB
Optimizer stripped from runs/segment/train/weights/best.pt, 54.0MB

Validating runs/segment/train/weights/best.pt...
Ultralytics YOLOv8.1.37 # Python 3.10.12 torch 2.2.1+cu121 CUDA:0 (Tesla T4, 15102MiB)
YOLOv8-seg summary (Fused): 245 layers, 2725279 parameters, 0 gradients, 139.0 GiB

```

Class	Images	Instances	BoxP	P	AP ₅₀	AP ₅₀₋₉₅	Recall	F	map50	map50-95	Speed A/s [inference, train]
all	50	115	0.961	0.911	0.954	0.912	0.941	0.895	0.931	0.900	
crack	50	28	0.963	0.95	0.977	0.960	0.941	0.95	0.937	0.927	
lack of fusion	50	28	0.964	1	0.973	0.972	0.942	1	0.933	0.930	
overlap	50	28	0.963	1	0.965	0.972	0.967	1	0.950	0.950	
rust	50	11	0.970	0.965	0.976	0.96	0.965	0.975	0.951	0.950	
slag inclusion	50	28	0.960	1	0.972	0.971	0.937	0.95	0.94	0.925	

```

Speed: 0.42s preprocess, 18.7ms inference, 0.8ms loss, 4.7ms postprocess per image
Results saved to runs/segment/train

```

Figure 4.9: Completion of training of model

The created YOLO V8 segmentation model can be deployed for various uses. One of the ways is to deploy it locally, running the model directly on our machine or device. This entails installing the necessary software, loading the trained model, and conducting inference on images or videos locally. Alternatively, we can opt for cloud deployment, leveraging platforms like Amazon Web Services (AWS), Google Cloud Platform (GCP), or Microsoft Azure. With this method, we can set up virtual machines or containers in the cloud, install the required dependencies, and deploy the model for inference. Cloud deployment offers scalability and flexibility, allowing us to adjust resources as needed. Edge deployment is another possibility, involving deploying the model on edge devices such as Raspberry Pi or mobile devices. This requires optimizing the model for resource-constrained environments. Web deployment is another avenue, where you integrate the YOLO model into web applications or services. This typically involves creating APIs or web services to serve model predictions over HTTP requests, enabling users to access predictions through web browsers or integrate them into web-based applications. Each of these deployment method offers its own advantages. The model trained has various capabilities and the usage of it to create a GUI using python is explored here.

CHAPTER 5

RESULTS AND DISCUSSION

5.1 TRAINING RESULTS OF THE DEFECT DETECTION MODEL

The defect detection model underwent successful training using an image dataset comprising 1634 images, which were further categorized into training (1484), validation (99), and testing (51) sets. The training process spanned across 20 epochs, ensuring comprehensive coverage of the dataset and facilitating improved model accuracy over iterations. The evaluation metrics yielded insightful results, including Precision (P), which gauges the accuracy of positive predictions, and Recall (R), indicative of the model's capability to correctly identify positive instances. Additionally, the mean Average Precision (mAP) scores, particularly at IoU thresholds of 50% and across a range from 50% to 95%, provided comprehensive insights into the model's accuracy levels across varying conditions. These performance indicators serve as vital benchmarks for assessing the effectiveness and robustness of the defect detection model. Leveraging these metrics, the following confusion matrix and visualization of training and validation set performances through graphical representations are developed, aiding in further analysis and refinement of the model.

Table 5.1: Training results of accuracies for each Epoch

Epoch	Images	Box				Mask				
		Instances	P	R	mAP50	mAP50-95	P	R	mAP50	mAP50-95
1	99	113	0.934	0.582	0.703	0.347	0.835	0.51	0.587	0.212
2	99	113	0.547	0.611	0.657	0.314	0.501	0.559	0.603	0.214
3	99	113	0.611	0.36	0.406	0.246	0.611	0.36	0.408	0.196
4	99	113	0.646	0.628	0.731	0.392	0.77	0.512	0.665	0.275
5	99	113	0.811	0.738	0.747	0.413	0.753	0.696	0.68	0.325
6	99	113	0.677	0.515	0.622	0.322	0.548	0.45	0.511	0.201
7	99	113	0.668	0.664	0.715	0.405	0.613	0.641	0.667	0.261
8	99	113	0.921	0.716	0.813	0.452	0.863	0.658	0.739	0.399
9	99	113	0.88	0.845	0.912	0.512	0.862	0.829	0.881	0.375
10	99	113	0.938	0.891	0.941	0.559	0.91	0.849	0.869	0.456
11	99	113	0.87	0.928	0.917	0.571	0.842	0.869	0.864	0.453
12	99	113	0.878	0.899	0.94	0.536	0.823	0.844	0.855	0.425
13	99	113	0.927	0.855	0.938	0.55	0.882	0.843	0.887	0.412
14	99	113	0.972	0.832	0.955	0.555	0.898	0.785	0.888	0.459
15	99	113	0.952	0.848	0.943	0.577	0.894	0.818	0.896	0.472
16	99	113	0.939	0.88	0.952	0.593	0.884	0.856	0.906	0.454
17	99	113	0.896	0.875	0.939	0.566	0.853	0.823	0.857	0.442
18	99	113	0.964	0.92	0.955	0.593	0.907	0.874	0.903	0.449
19	99	113	0.976	0.907	0.961	0.585	0.948	0.889	0.921	0.497
20	99	113	0.948	0.916	0.966	0.597	0.904	0.891	0.923	0.476

Precision (P) represents the accuracy of the model's positive predictions, calculated as the ratio of correctly predicted positive instances (true positives) to the total instances predicted as positive (true positives + false positives). A higher precision indicates fewer false positive predictions, signifying better accuracy in identifying positive instances. The model has the lowest value of 0.547 and has its final scores at the average of 9 showing that it has high precision.

Recall (R), also known as sensitivity or true positive rate, measures the model's ability to capture all actual positive instances in the dataset. It is calculated as the ratio of correctly predicted positive instances (true positives) to the total actual positive instances (true positives + false negatives). A higher recall indicates that the model effectively captures most of the positive instances present in the dataset. The model

has the lowest score of 0.3 at start and by the end of the training the model achieves a score of 0.9 showing that the predictions made were accurate.

Mean Average Precision at an IoU threshold of 0.5 (mAP50) evaluates the model's precision-recall trade-off across different object classes. It calculates the average precision across classes at an Intersection over Union (IoU) threshold of 0.5, where IoU measures the overlap between predicted bounding boxes and ground truth bounding boxes. mAP50 provides an overall measure of the model's performance in detecting objects with a moderate level of overlap with ground truth annotations. The model has the lowest value of 0.4 and by the end of the training its able to differentiate more easily giving a result of 0.9 at the end. Mean Average Precision across IoU thresholds from 0.5 to 0.95 (mAP50-95) assesses the model's performance across a range of IoU thresholds, providing a broader evaluation of its accuracy. It calculates the average precision by averaging precision values obtained at IoU thresholds ranging from 0.5 to 0.95, indicating the model's ability to detect objects accurately under various degrees of overlap criteria.

5.1.1 Confusion matrix of the Defect Detection Model

A confusion matrix serves as a crucial instrument for evaluating the effectiveness of a machine learning model, particularly in classification assignments. It offers a comprehensive breakdown of the model's predictions along with the actual results. Within the matrix, True Positives (TP) denote instances accurately forecasted as positive, False Positives (FP) indicate inaccurately forecasted positives, True Negatives (TN) represent correctly anticipated negatives, and False Negatives signify erroneously predicted negatives. The YOLO confusion matrix, as seen in Figure 5.1 offers valuable insights into the object detection model's performance across different classes, including evaluating class-specific performance, identifying misclassification patterns, assessing overall accuracy, detecting class imbalance issues, and recognizing performance discrepancies. Analyzing off-diagonal elements reveals recurring misclassification patterns, guiding model refinement strategies. Despite overall accuracy assessment, it's crucial to consider class imbalances' impact. Disproportionate class representation signals potential imbalance problems, impacting precision and recall, with mitigation strategies like data augmentation. Discrepancies

in precision and recall highlight areas for improvement, addressed through adjustments in thresholds and additional training data. Ultimately, the YOLO confusion matrix guides enhancements for improved accuracy and robustness across classes.



Figure 5.1: Confusion matrix

We can see that the model was able to accurately detect the defects for every class successfully like for cracks, lack of fusion slag inclusion and overlap by detecting 20 instances out of the 20 validation images. The model was not able to accurately detect for porosity but was able to detect 23 instances out of 33 validation images. The model gave us the best results for overlap by detecting it accurately with 20 TP and 107 TN. The model can be improved by providing more images for the other 4 defects especially for porosity. Table 5.2 provides a detailed view of the values Tp, Tn, Fp and Fn.

Table 5.2: Values obtained from the Confusion Matrix

Class	Tp	Tn	Fn	Fp
Crack	20	105	0	2
Lack of Fusion	20	102	0	5
Overlap	20	107	0	0
Pore	23	88	10	6
Slag inclusion	20	106	0	1

5.1.2 Model evaluation based on Loss Functions

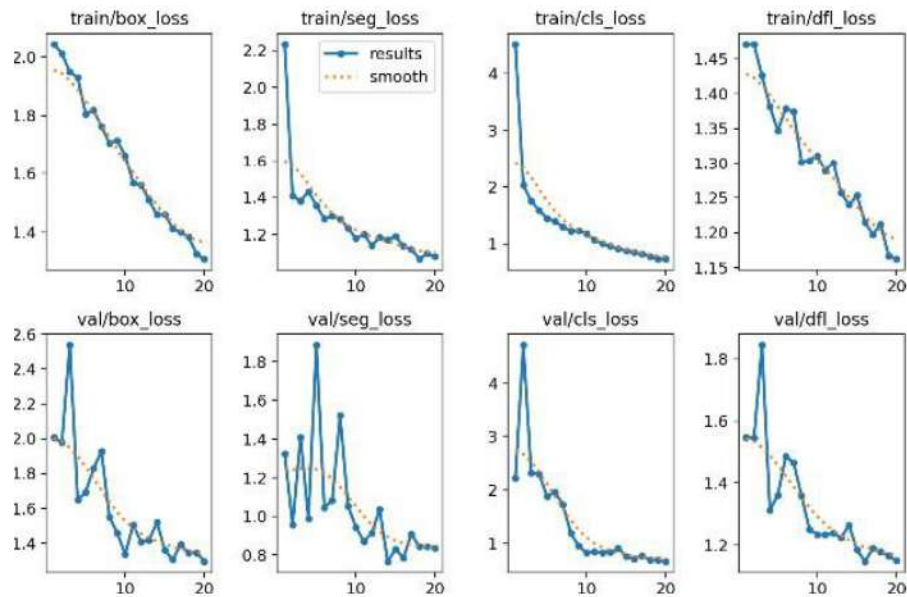


Figure 5.2: Training and Validation Losses of the model

The model was trained upon 1,635 images and the trained results were plotted into these graphs. Figure 5.2 discusses the Training and Validation loss graphs obtained from YOLO. The X axis consists of the Epochs ranging from 0 — 20 and the Y axis consists of the magnitudes of those specific graphs. The graph can be majorly categorized into two Train and Validation losses. Both categories contain 4 sets of values which are Box loss, Segmentation loss, Class loss and Differential loss. These

loss gives insight upon the ability of the model. We observe a uniform trend of the loss decreasing showing that the increase in number of epochs results in a decrease in the loss thus making the model better suitable for this purpose.

5.1.3 Model evaluation based on Metrics

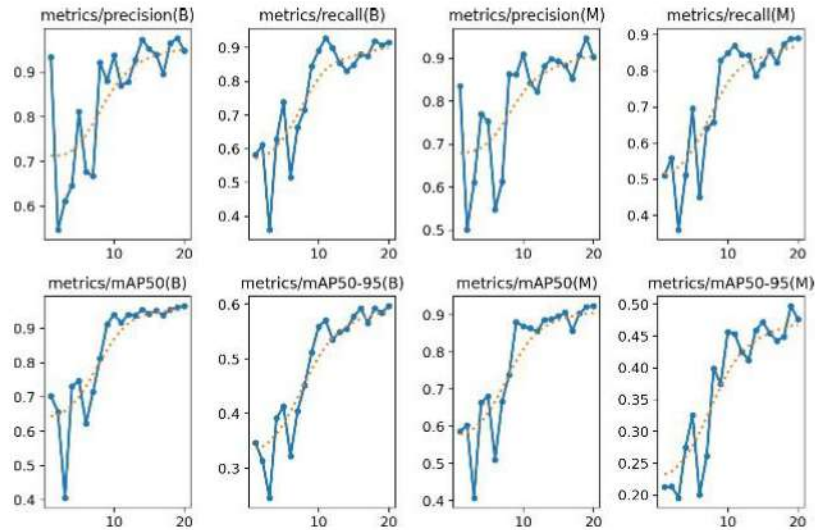


Figure 5.3: Metrics of the model bases on Bounding Boxes and Maps

The model was trained upon 1,635 images and the trained results were plotted into these graphs. Figure 5.3 discusses the Metrics graph obtained from YOLO .The X axis consists of the Epochs ranging from 0 — 20 and the Y axis consists of the magnitudes of those specific graphs. The graph can be majorly categorized into two Box based metrics and Map based metrics. Both categories contain 4 sets of values which are precision, recall, mAP50 and mAP(50-95). These metrics gives insight upon the ability of the model to detect the defects accurately. Weobserve a uniform trend of the metrics increasing showing that the increase in number of epochs results in a decrease in the loss thus making the model better suitable for detecting defects accurately.

5.2 ACCURACY OF THE PREDICTED IMAGES OF THE MODEL

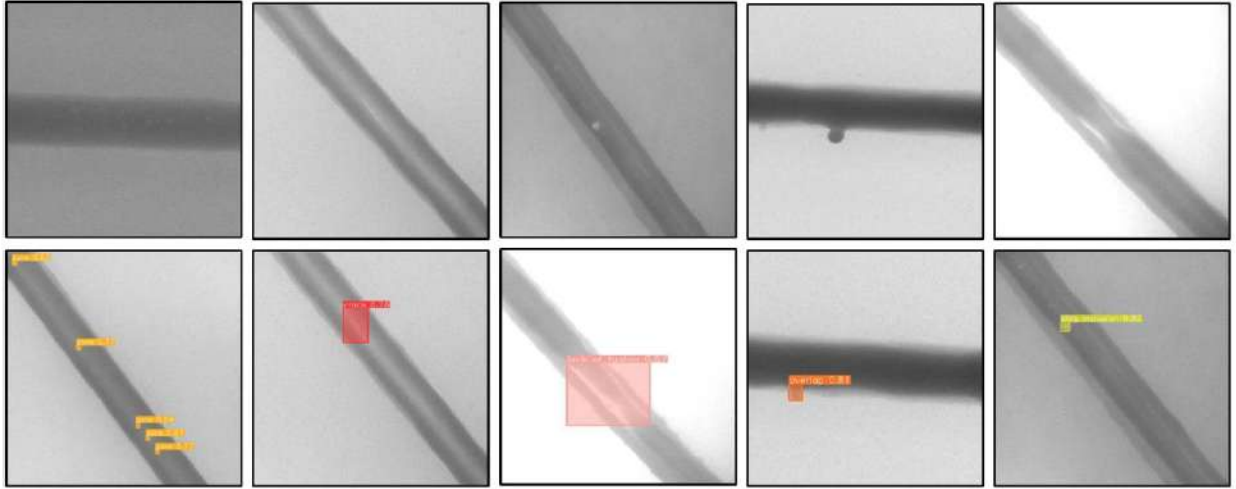


Figure 5.4: Images predicted by the YOLO v8 model

The predicted images as shown in Figure 5.4 give us a clear idea on the confidence scores of the images. The images clearly show us the location of the defect and their confidence scores. The prediction of these images shows that the confidence score for porosity has a huge variation from 0.5 to 0. The confidence scores for cracks have an average of 0.8 and they were able to be detected accurately. Overlap defects had the most accurate predictions averaging at 0.9 and were accurately detected. Slag inclusions were also detected with a confidence score averaging at 0.8 and has accurate predictions. The addition of more images would increase these confidence scores and more clear annotations will help in improving the performance of the model.

5.3 PREDICTING IMAGES IN REAL TIME

YOLO is highly compatible with Python, enabling us to conduct predictions in images and videos. YOLO offers pre-trained models trained on extensive datasets like COCO (Common Objects in Context) or VOC (Visual Object Classes). These pre-trained models can seamlessly be imported into Python using libraries such as OpenCV or PyTorch. Once imported, these models are capable of detecting objects in images or videos. OpenCV serves as a potent computer vision library in Python, supporting YOLO integration. By integrating YOLO with OpenCV, object detection can be performed on images and videos. OpenCV provides functionalities to load the YOLO model, conduct inference, and delineate bounding boxes around detected objects.

YOLO is constructed using deep learning frameworks like TensorFlow and PyTorch. These frameworks can be harnessed in Python to train custom YOLO models on specific datasets or fine-tune pre-trained models for targeted tasks. These frameworks furnish APIs for loading pre-trained YOLO models, executing inference, and tailoring the model architecture to meet specific requirements. Python boasts a rich array of libraries and tools for crafting custom applications employing YOLO. Interactive applications or web services leveraging YOLO for object detection can be developed. Python's user-friendly nature and adaptability streamline the integration of YOLO into diverse applications and workflows. Python thus presents a user-friendly and adaptable environment for employing YOLO in object detection tasks. Whether working with pre-trained models or developing custom applications, Python's extensive library ecosystem simplifies the integration of YOLO into various projects.

For example, the model can be directly tested upon images by downloading the required libraries Ultralytics, PIL and cv2. The Ultralytics library provides implementations for various deep learning models, including YOLO (You Only Look Once), which is a popular object detection algorithm. The YOLO class imported from Ultralytics allows you to use pre-trained YOLO models for tasks such as object detection in images and videos. PIL (Python Imaging Library) PIL is a widely used library for image processing tasks in Python. The Image module imported from PIL provides functions and classes for creating, opening, manipulating, and saving images in various formats. It's commonly used for tasks like image loading, resizing, cropping, and basic image manipulation. cv2 (OpenCV) cv2 is the Python binding for OpenCV (Open-Source Computer Vision Library), which is a powerful library for computer vision tasks. OpenCV provides a wide range of functionalities for image and video processing, including image manipulation, feature detection, object recognition, and more. It's often used for tasks like reading and writing images and videos, image transformation, feature extraction, and object detection. Thus, the Ultralytics library provides access to deep learning models like YOLO for object detection tasks, while PIL and cv2 are popular libraries for image processing and computer vision tasks in Python. This combination of libraries allows us to leverage pre-trained deep learning models for object detection while also performing various image processing operations.

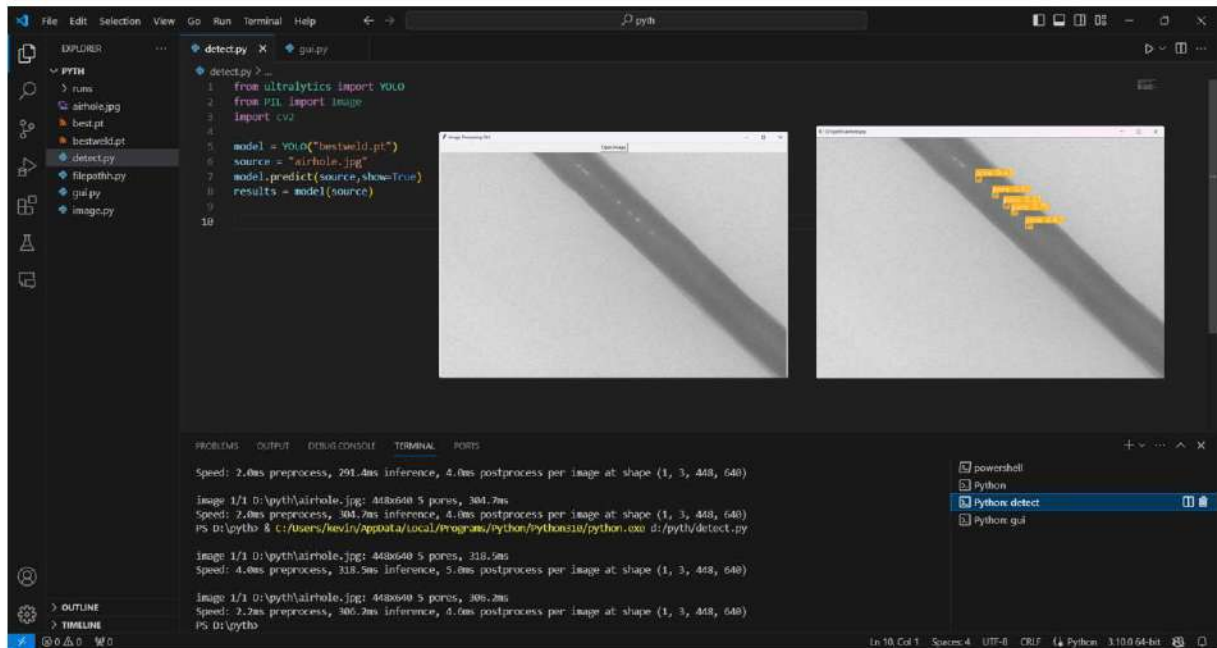


Figure 5.5: Python program created using the YOLO model

One such implementation can be seen here in Figure 5.5. Here the required libraries are first imported, they are YOLO from Ultralytics, Images from PIL and cv2. Now the model is specified to be “bestweld.pt”. “bestweld.pt” as obtained as a training weight during the training of the model in Google Collab. The weight is downloaded and saved in the same file as the python program file. A custom image, as seen in Figure 5.6, is also stored in the same directory as the Python program file. The name of the custom file is then used in the program to predict images, as shown in Figure 5.7, in real time on new thermal weld images. The images are then displayed and saved in the directory.

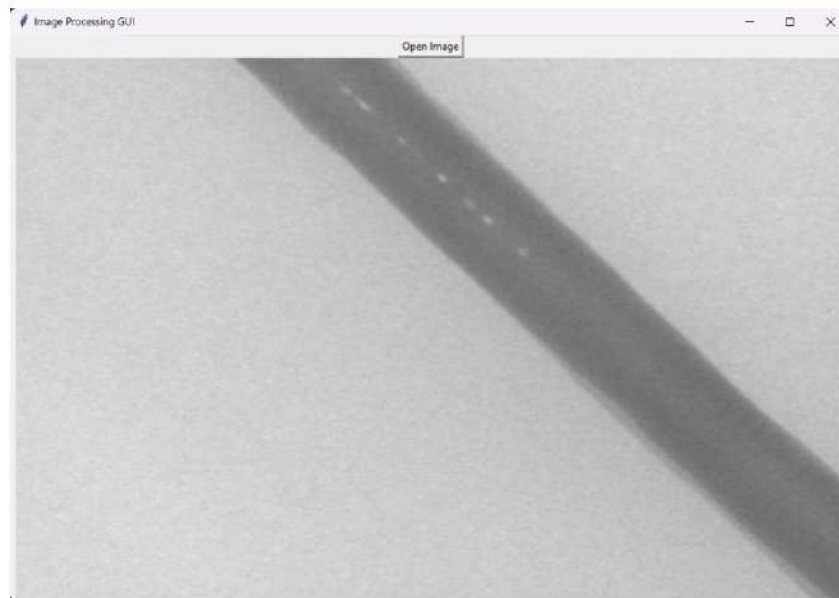


Figure 5.6: Image opened using the python program

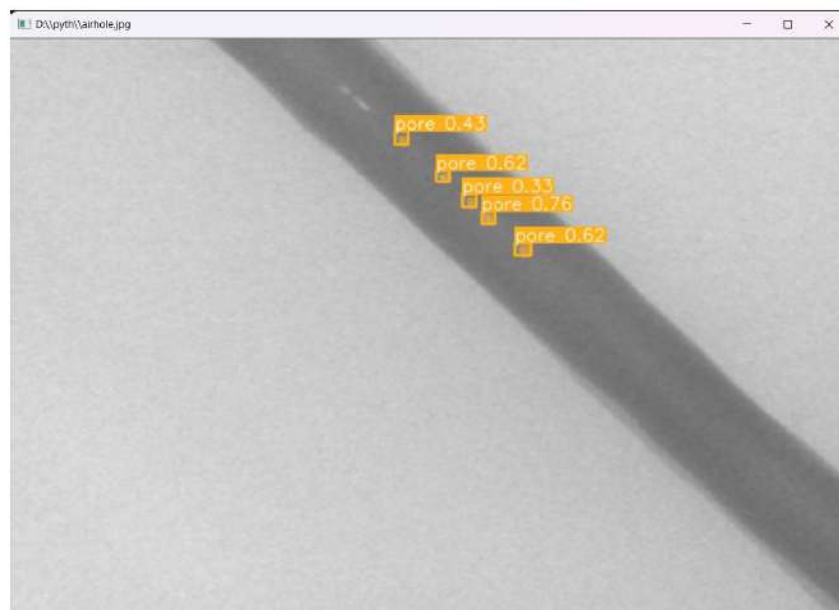


Figure 5.7: Prediction made on the image using the python program

CHAPTER 6

SUMMARY AND CONCLUSION

This research work proposes a defect detection system which is in compliance to the recent trends. The combination of semantic segmentation and thermal imaging provides a new scope and increases the efficiency and accuracy of the system. The Lepton 1R infrared camera used to capture thermal images has a noise equivalent temperature difference of 50 mK, resolution of 160 x 120 pixels, a maximum sampling rate of 7 Hz, and a spectral range of 8 to 14 μ m. The image obtained from this is used in the deep learning model with the help of raspberry pi which helps us to communicate the image to the computer system. The system uses YOLO V8, which is a recently developed deep learning model known for its exceptional accuracy and speed. Integration of YOLO V8 gives us the accuracy and speed which is desired in the thermal imaging and semantic segmentation systems. The model trained with the help of YOLO V8 generally show high efficiency compared to the other models. The system proposed is able to detect the thermal defects using semantic segmentation and thus gives a clear identification of the defect and its location. The system was also able to detect defects in real time when the model is trained with images produced for that specific kind of welded metals and is trained based on historic data of that welded part. YOLO V8 also generates detailed graphs on the trained model and its ability to detect defects which can be used for further research and to increase the performance and efficiency of this system.

The research shows us how the thermal imaging systems can be further enhanced with the help of deep learning models like YOLO v8 to detect defects in the thermal images. The developed model was based on open access image dataset obtained from KAGGLE. Which was annotated to suit the research. The images obtained from the Raspberry Pi and Lepton 1r setup was used to detect defects in real time allowing us to prove that the setup can be used to detect defects on completely new data.

Further the images obtained can be used to develop the model further allowing us to improve the defect detection model to give better accuracies. The dataset of 1634 images divided into train(1484), valid(99) and test(51) trained through 20 epochs gave an average of 0.9 accuracies tested upon the images belonging to this dataset but

unseen by the model. Further upon the image dataset can be further enhanced for the model to give the same accuracies over completely new images obtained using a thermal camera.

Similar setup can be used in an industrial scale by providing access to historical data available in that specific industry and training a Yolo v8 model specific to that database. For example, the same setup can be used by replacing the Lepton 1r with better cameras from the FLIR A-series would help in improving the accuracies of the detected defects and improve the detection range of the defect. Thus, the research can be further extended to suit the Industrial needs and provide a proper defect detection system based upon Semantic segmentation and Thermal Imaging. A separate user interface can be created to show the detected defects giving its confidence score and the location of the defects to enable easy and simple interactions for the end user to navigate through and view the images taken by the setup and further information that can be gained from the Defect detection model.

REFERENCES

- [1] Afify, H. & Mohammed, Kamel K. & Hassanien, Aboul Ella, "An improved framework for polyp image segmentation based on SegNet architecture", International Journal of Imaging Systems and Technology, (2021)31. 10.1002/ima.22568.
- [2] Ajay Shrestha and Ausif Mahmood "Review of Deep Learning Algorithms and Architectures", (2019) 10.1109/ACCESS.2019.2912200.
- [3] Albawi, Saad & Abed Mohammed, Tareq & ALZAWI, Saad, "Understanding of a Convolutional Neural Network", (2017), 10.1109/ICEngTechnol.2017.8308186.
- [4] Ali Sarhadi, Rodrigo Q. Albuquerque, Martin Demleitner, Holger Ruckdäschel, Martin A. Eder, "Machine learning based thermal imaging damage detection in glass-epoxy composite materials, Composite Structures", Volume 295, 115786, ISSN 0263-8223, (2022), <https://doi.org/10.1016/j.compstruct.2022.115786>.
- [5] Angela Busheska, Nara Almeida, Nicholas Sabella, Eudes de A. Rocha, "Machine Learning and Thermography Applied to the Detection and Classification of Cracks in Building", (2022), <https://doi.org/10.48550/arXiv.2212.14730>.
- [6] Antonio, Luiz & Medeiros, Heitor & Macêdo, David & Zanchettin, Cleber & Oliveira, Adriano & Ludermir, Teresa, "SegNetRes-CRF: A Deep Convolutional Encoder-Decoder Architecture for Semantic Image Segmentation", (2018), 1-6. 10.1109/IJCNN.2018.8489376.
- [7] Bang, Hyun-Tae & Park, Solmoi & Jeon Haemin. (2020), "Defect identification in composite materials via thermography and deep learning techniques. Composite Structures", 246. 112405. (2021) 10.1016/j.compstruct.2020.112405.
- [8] Charvát Michal, Bc. System for people detection and localization using thermal imaging cameras. 2020. [Doi.org/3390/7348199](https://doi.org/10.3390/7348199)
- [9] Chitradevi, B. and P.Srimathi, "An Overview on Image Processing Techniques", International Journal of Innovative Research in Computer and Communication Engineering 2, (2014), 6466-6472.
- [10] Dipali Bhabad, Surabhi Kadam, Tejal Malode, Girija Shinde and Dipak Bage, Object Detection for Night Vision using Deep Learning Algorithms. 2023, <https://doi.org/10.14445/22312803/IJCTT-V7I12P11>
- [11] Dražanský M, M Charvát, I Macek, J Mohelníková. Thermal Imaging Detection System: A Case Study for Indoor Environments. September 2023 / DOI:3390/s2318782

- [12] Fang, Qiang & Mamoudan, Farima & Maldague, Xavier, "A Method of DefectDepth Estimation for Simulated Infrared Thermography Data with Deep Learning", Applied Sciences, 10. 6819, (2020), 10.3390/app10196819.
- [13] Haodong Zhang, Zuzhi Chen and Chaoqun Zhang. Weld Defect DetectionBased on Deep Learning Method. 201 DOI: 1109/COASE.201884299
- [14] Hartley, R. Multiple View Geometry in Computer Vision. 2nd ed. Cambridge University Press, 200 ISBN 354049698X.
- [15] Heba M. Afify, Kamel K. Mohammed and Aboul Ella Hassanien, An improved framework for polyp image segmentation based on SegNet architecture. 2021, <https://doi.org/1002/ima.2256>
- [16] Henderson, G. Wiring Pi- GPIO Interface library for the Raspberry Pi. 2020. [Online; Accessed: 10-01-2020]. Available at: <http://wiringpi.com/>.
- [17] Herrmann, C., Ruf, M. and Beyerer, J. CNN-based thermal infrared person detection by domain adaptation. Autonomous Systems: Sensors, Vehicles, Security, and the Internet of Everything. International Society for Optics and Photonics. 201 p. 106430
- [18] Hyun-Tae Bang, Solmoi Park and Haemin Jeon. Defect identification in composite materials via thermography and deep learning techniques. 2020, DOI:1016/j.compstruct.2020.11240
- [19] Ivanov, B., Ruser, H. and Kellner, M. Presence detection and person identification in Smart Homes. [Neubiberg, University of Bundeswehr Munich and Passau, FORWISS, University Passau]. Jul 201
- [20] John, Vijay & Mita, Seichi. "Deep Feature-Level Sensor Fusion Using Skip Connections for Real-Time Object Detection in Autonomous Driving", Electronics, 10, (2021) 10.3390/electronics10040424.
- [21] Juan, Liu & Li, Chao & Wu, Xiaojun & Palade, Vasile & Fang, Wei , "An Effective Method of Weld Defect Detection and Classification Based on Machine Vision", IEEE Transactions on Industrial Informatics, PP. 1-1. (2019), 10.1109/TII.2019.2896357.
- [22] Kamal, A. YOLO, YOLOv2 and YOLOv3: All You want to know [Medium.com – online publishing platform]. 201 [Online; Accessed: 18-06-2020]. Available at: https://medium.com/@amrokamal_47691/yolo-yolov2-and-yolov3-all-you-want-to-know-7e3e92dc489
- [23] Krizhevsky, A., Sutskever, I. and Hinton, G. E. ImageNet Classification with Deep Convolutional Neural Networks. In: Pereira, F., Burges, C. J. C., Bottou,

- [24] L. and Weinberger, K. Q., ed. *Advances in Neural Information Processing Systems* 2 Curran Associates, Inc., 2012, p. 1097–110 Available at: <http://papers.nips.cc/paper/4824-imagenet-classification-with-deepconvolutional-neural-networks.pdf.0/>
- [25] Kyle Simek. *Dissecting the Camera Matrix: Part 1,2,3 – Extrinsic/Intrinsic Camera Matrix* [Kyle Simek's Computer Vision Blog]. 2012-201 [Online; Accessed: 18-06-2020]. Available at: <http://ksimek.github.io/2012/08/14/decompose/>
- [26] Lei Yang, Huaixin Wang, Benyan Huo, Fangyuan Li, Yanhong Liu, "An automatic welding defect location algorithm based on deep learning", *NDT & E International*, Volume 120, 102435, ISSN 0963-8695, (2021), <https://doi.org/10.1016/j.ndteint.2021.102435>.
- [27] Li, Yue & Zhang, Xueting & Shen, Zhangyi. "YOLO-Submarine Cable: An Improved YOLO-V3 Network for Object Detection on Submarine Cable Images", *Journal of Marine Science and Engineering*, 10. 1143, (2022). [10.3390/jmse10081143](https://doi.org/10.3390/jmse10081143).
- [28] Luiz Antonio de Oliveira Junior, Heitor R. Medeiros, David Macedo and Cleber Zanchettin, *SegNetRes-CRF: A Deep Convolutional Encoder-Decoder Architecture for Semantic Image Segmentation*. 2017, DOI: 1109/TPAMI.201264461
- [29] Marius Popovici, "Active thermography – Non-destructive & Non-contact material testing", *Progress of Cryogenics and Isotopes Separation* Vol. 15, Issue 2 (2012)
- [30] Muhammad Hussain, *YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection*, 202 DOI: [10.3390/machines1107067](https://doi.org/10.3390/machines1107067)
- [31] Muhammad Hussain, "YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection", (2023), <https://doi.org/10.3390/machines11070677>.
- [32] Pei-Fen Tsai, Chia-Hung Liao and Shyan-ming yuan. *Using Deep Learning with Thermal Imaging for Human Detection in Heavy Smoke Scenarios*. 2022; <https://doi.org/10.3390/s2214535>
- [33] Priya Aggarwal, Narendra Kumar Mishra, Binish Fatimah, Pushpendra Singh, Anubha Gupta, Shiv Dutt Joshi, "COVID-19 image classification using deep learning: Advances, challenges and opportunities, *Computers in Biology and Medicine*", Volume 144, 105350, ISSN 0010-4825, (2022), <https://doi.org/10.1016/j.compbiomed.2022.105350>.
- [34] Roberto Marani, Davide Palumbo, Umberto Galietti, Tiziana D'Orazio, "Deep

learning for defect characterization in composite laminates inspected by step-heating thermography, *Optics and Lasers in Engineering*, Volume 145, 106679, ISSN 0143-8166, (2021), <https://doi.org/10.1016/j.optlaseng.2021.106679>.

- [35] Saad ALBAWI, Tareq Abed MOHAMMED and Saad AL-ZAWI, Understanding of a Convolutional Neural Network. 2017, DOI: 1109/ICEngTechnol.201830818
- [36] Wael M. Elmessery, Joaquín Gutiérrez and Gomaa G. Abd El-Wahhab. YOLO-Based Model for Automatic Detection of Broiler Pathological Phenomena through Visual and Thermal Images in Intensive Poultry Houses. 202 <https://doi.org/3390/agriculture1308152>
- [37] Wael M. Elmessery, Joaquín Gutiérrez, Gomaa G. Abd El-Wahhab, Ibrahim A. Elkhayat, Ibrahim S. El-Soaly, Sadeq K. Alhag, Laila A. Al-Shuraym, Mohamed
- [38] Akela, Farahat S. Moghanm and Mohamed F. Abdelshafie, "YOLO-Based Model for Automatic Detection of Broiler Pathological Phenomena through Visual and Thermal Images in Intensive Poultry Houses", *Agriculture* (2023), 13, 1527. <https://doi.org/10.3390/agriculture13081527>.
- [39] Yunze He, Baoyuan Deng, Hongjin Wang, Liang Cheng, Ke Zhou, Siyuan Cai, Francesco Ciampa, "Infrared machine vision and infrared thermography with deep learning: A review", *Applied ELSEVIER, Infrared Physics & Technology*, (2021), <https://doi.org/10.1016/j.infrared.2021.103754>.
- [40] Zhang, Haodong & Chen, Zuzhi & Zhang, Chaoqun & Xi, Juntong & Le, Xinyi, "Weld Defect Detection Based on Deep Learning Method", 1574-1579, (2019), 10.1109/coase.2019.8842998.

ANNEXURE

First page of Review paper

**1st International Conference on Advancement in Energy (ऊर्जाSangam-2023)
18th –20thDecember, 2023
MNNIT Allahabad, Prayagraj, Uttar Pradesh, India**

THERMAL IMAGING: DETECTING DEFECT WITH SEMANTIC SEGMENTATION

J R Deepak ^{a,*}, Chidambaresh N^a, Kevin Rajadurai E ^a

*School of Mechanical Engineering,
Sathyabama Institute of Science and Technology, Chennai
Tamil Nadu, India.*

**E-mail: Deepak.prod@sathyabama.ac.in*

ABSTRACT

Thermal imaging systems are high-quality system known for non-invasive, high-resolution defect identification. The infrared radiation emissions are captured by these thermal imaging systems to carry out meticulous inspections without any physical contact. The thermal images captured is of high contrast, in a color scale which maps the heat signatures of the object that is inspected, thereby differentiating the temperature variations with respect to that color scale. But these thermal images generated in the form of RGB images do not indicate accurate temperature values. To examine these temperature variations, addition of deep learning algorithm a subfield of machine learning, such as semantic segmentation assistance to categorize these thermal images into categories in the pixel level. YOLOv8 is a deep learning model which is designed for real time object segmentation, making them suitable for applications that require quick and accurate identification of features of the image. When integrated with thermal imaging systems, these YOLOv8 modules create a real breakthrough in the field of Non-Destructive Testing (NDT).

In this research work, a thermal imaging defect detection system with semantic segmentation is developed to detect various defects in the thermal images of the welded metal sheets. The thermal imaging defect detection system includes an induction coil to preheat the welded metal sheet, Lepton 3.1R camera module to generate thermal image and a Raspberry Pi 4 for transmitting the image to the computer in which the YOLOv8 module is executed. Rob-flow software pre-processes and annotates the thermal image into labels. These labeled images are categorized and subsequently fed into the YOLOv8 module for model training which enables the system to detect the labeled defects. The fusion of thermal imaging with deep learning such as the YOLOv8 model in this research elevates the field of inspection (NDT) by enabling automated, high-speed analysis of vast thermal datasets which enhances defect detection accuracy and its utility in the field of non-destructive testing (NDT).

Keywords: Thermography, Deep Learning, Semantic Segmentation, YOLO V8.

1. INTRODUCTION

Thermal imaging systems have gained widespread recognition for their ability to deliver high-quality results in the field of non-invasive, high-resolution defect identification. These advanced systems harness the power of infrared radiation emissions to conduct meticulous inspections without the need for any physical contact. The output from these systems is a set of thermal images of high contrast. These images generated are of a certain color scale, enabling the visual mapping of heat signatures across the inspected object. In doing so, they

ACCEPTANCE MAIL OF REVIEW PAPER

3/30/24, 8:26 AM

Gmail - Status of your Full-Length Paper: Accepted for Oral Presentation in 'First International Conference on Advancement in Ene...



Kevin Rajadurai <kevinrajadurai@gmail.com>

Status of your Full-Length Paper: Accepted for Oral Presentation in 'First International Conference on Advancement in Energy (ऊर्जाSangam-2023)'

1 message

Microsoft CMT <email@msr-cmt.org>
Reply-To: Rahul Dev <rahuldsurya@mnnit.ac.in>
To: Kevin E Rajadurai <kevinrajadurai@gmail.com>

Fri, Dec 1, 2023 at 1:54 PM

Dear Author(s),

Thank you for submitting your paper in 'First International Conference on Advancement in Energy (ऊर्जाSangam-2023)' going to be held from December 18, 2023 to December 20, 2023 in Mechanical Engineering Department, Motilal Nehru National Institute of Technology Allahabad, Prayagraj, Uttar Pradesh, India.

We are pleased to inform you that your Full-length Paper (Paper ID 288) titled "Thermal Imaging: Detecting defect with Semantic segmentation" authored by Kevin Rajadurai, Riswanth S has been accepted for Oral Presentation.

The registration is Open now, You are requested to register (if not registered yet) as early as possible by following the steps:

(i) The Payment of Registration Fee can be done by using the link : <https://www.onlinesbi.sbi/sbicollect/collecthome.htm?corpID=5483621> . During payment, please do not forget to enter the Unique ID : 08/ICAE/MED/2023-24/SNFCE, in the Remarks section.

(ii). Thereafter, You have to register for the participation in the conference by using the google form: https://docs.google.com/forms/d/e/1FAIpQLSeSHaOe4_rIHPOjnG95g6oyGvXVBSlzz4kwEUOv7cGwY90w9g/viewform?pli=1

The schedule of your paper will be informed soon. We are looking forward to seeing you participating in the conference.

Thank you.

Regards.

.....

Conference Chair(s)

First International Conference on Advancement in Energy (ऊर्जाSangam-2023)
MNNIT Allahabad.

To stop receiving conference emails, you can check the 'Do not send me conference email' box from your User Profile.

Microsoft respects your privacy. To learn more, please read our [Privacy Statement](#).

Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

<https://mail.google.com/mail/u/0/?ik=a2212f9e53&view=pt&search=all&permthid=thread-f:1784067229486984194&simpl=msg-f:1784067229486984194> 1/1

1st International Conference on Advancement in Energy



ऊर्जाSangam-2023

18th-20th December 2023

Organized by

Department of Mechanical Engineering

Motilal Nehru National Institute of Technology Allahabad, Prayagraj, (U.P.) INDIA

In Collaboration with:



Sponsored by:



Published by:



Certificate of Presentation

This is to certify that Mr./Ms. **CHIDAMBARESH N.** has

Participated/Presented his/her Paper titled **"Thermal Imaging: Detecting defect with Semantic segmentation"**

(**Paper ID: 288**) Authored by **J R Deepak Chidambaresh N.**

Kevin Rajadurai E in the **1st International Conference on**

Advancement in Energy ऊर्जाSangam-2023 held during **18-20 December 2023.**

Prof. Ravi Prakash
Conference Chair

Prof. K.N. Pandey
Head, MED



FIRST PAGE OF EXPERIMENTAL PAPER

WELD DEFECT DETECTION USING THERMAL SCANNING AND SEMANTIC SEGMENTATION: AN MODERN APPROACH ON NON-DESTRUCTIVE TESTING (NDT)

J R Deepak^{*1}, Kevin Rajadurai E¹, Chidambaresh N¹

*School of Mechanical Engineering, Sathyabama Institute of Science and Technology,
Chennai, India*

*deepak.prod@sathyabama.ac.in
kevinrajadurai@gmail.com
chidambareshn@gmail.com*

ABSTRACT

Welding has been integral to every industry globally and serves as the backbone in all major aspects like in aerospace, automotive, construction and energy industries. Timely rectification of defects not only reduces expenses related to rework, material wastage, and downtime but also maintains high-quality standards. To prevent these the welded components are inspected using various Non-destructive techniques. NDT helps us to detect these defects without having to destroy or damage the weld and its surroundings. Thermal imaging, among other NDT has proven to be the best method to detect weld defects due to several key reasons. They capture temperature differences effectively, making defects stand out against the background where they exhibit distinct thermal characteristics. It also ensures swift inspections without altering the weld or risking material damage. Additionally, these images enable real-time monitoring during welding, providing immediate feedback for timely defect identification and corrective actions.

Thermal imaging, combined with deep learning algorithms like Semantic segmentation, allows for automated and high-speed defect detection, enhancing its accuracy. Semantic segmentation proves to be crucial for detecting weld defects, by providing detailed defect localization at the pixel level, enabling accurate assessment of defect severity. YOLOv8 segmentation module simplifies the training process and integrates seamlessly with other tools from Ultralytics, making it user-friendly for various applications. The images for the research were taken using a thermal imaging defect detection system incorporating Lepton 3.1R camera module, and Raspberry Pi 4. Utilizing Rob-flow software for pre-processing and annotation enhances defect detection efficiency. The defect detection model generated using Yolo V8 segmentation model was trained upon a dataset containing 2680 images (train 70%, valid 20%, test 10%) which were clearly annotated and augmented in Roboflow. The annotated images were used to create the Yolo V8 Segmentation model which was completed in 0.278 hrs with a preprocess speed of 0.4ms, 14.5ms inference, 0.0ms loss and 2.7ms postprocess per image.

Key words: NDT, Welding, Thermal imaging, Semantic segmentation, YOLO v8

1. INTRODUCTION

RE-2022-227310-plag-report

ORIGINALITY REPORT

18%

SIMILARITY INDEX

13%

INTERNET SOURCES

11%

PUBLICATIONS

9%

STUDENT PAPERS

PRIMARY SOURCES

1

sist.sathyabama.ac.in

Internet Source

1%

2

www.mdpi.com

Internet Source

1%

3

Submitted to Liverpool John Moores University

Student Paper

1%

4

www.coursehero.com

Internet Source

1%

5

fastercapital.com

Internet Source

1%

6

Aravamuthan, Srikanth. "Early Detection of Digital Dermatitis in Cattle Using Computer Vision", The University of Wisconsin - Madison, 2024

Publication

<1%

7

github.com

Internet Source

<1%

8

Submitted to University of Wollongong

Student Paper

<1%