Machine Learning and Computer Vision for Robotic Disassembly of E-Waste with Specific Emphasis on the Detection of Screws in Hard Drives

Pranav Bhatnagar

A thesis in partial fulfilment of the requirements
for the degree of
Bachelor of Computer Engineering

School of Computer Science and Engineering
University of New South Wales (UNSW) Sydney

April 2025

Abstract

In the context of saturated consumerism, shortened product lifespans, technological advancements and rapid generation of electronic waste across developed and developing nations, there is a need to evolve efficient, dynamic, secure, and widely implementable electronic waste management solutions. The focus of this project dives into the challenge of detecting screws on hard drives using a computer-vision-based algorithm capable of carrying out operations autonomously without the need for human coordination. Computer hard drive (HDD) disassembly was chosen due to its technological feasibility, economical viability and potential for resource and data recovery in the pre-processing stage and avoiding conventional data destruction methods like shredding or incineration. Through three different algorithms based on deep learning models such as You-Only-Look-Once, U-Net and Machine Learning model like Kmeans, we've compared and analysed their efficiency in classifying and detecting screws across multiple orientations and conditions in hard drives. An additional goal would be to make the algorithms further optimised through assessing them via various metrics such as Intersection Over Union, Recall, and Precision to ensure low latency and adaptability to real-life scenarios. Through computer vision and deep learning techniques, this thesis would strive to address challenges around hard drive disassembly and in turn, the environmental and data threatening impact of e-waste.

Keywords: Artificial Intelligence, Electronic Waste, Computer Vision, Hard Drives, Robotic Disassembly

Acknowledgement

I would like to express my utmost gratitude to my supervisor, Dr. Maurice Pagnucco, who very patiently guided me every step of the way and oriented my research toward a more productive direction. I would like to thank the University of New South Wales, for the continued opportunity to be present in this institution. I would like to further express my profound gratitude to the entire Faculty of Engineering for providing me with one of the most invigorating and rewarding years of my life. Lastly, I would like to express my heartfelt gratitude to my parents, my sister, and my friends for their unwavering and unconditional support throughout this journey.

Contents

List of Figures

List of Tables

Listings

Chapter 1

Introduction

Globally, as we reach a saturated point in terms of consumerism and production, it has become vital to address both the harmful impact of e-waste while considering its importance for resource recovery and finding rare earth materials. When it comes to e-waste management, an essential part of the term is its identification as 'waste', meaning it has no further use, is rejected, useless and excess to the owner in its current condition. E-waste refers to electrical or electronic equipment classified as waste and comprises all components, subassemblies, and consumables that are integrated into the equipment or product at the time it is classified as waste. E-waste becomes hazardous when it has substances such as mercury, lead, and brominated flame retardants. At the same time, e-waste also contains metals and materials of 'strategic importance' such as indium and palladium with the potential to include 60 different elements from the periodic table, including persistent organic pollutants or POPs. As revealed by the United Nations's fourth Global E-Waste Monitor, e-waste is rising five times faster than documented e-waste recycling. For the year 2022, only one quarter or 22.3 percent of the e-waste mass was recorded as systematically collected and recycled. The many reasons pointed out by the UN Global E-Waste Monitor include higher consumption patterns, decreased product cycles, design shortcomings, electronification, and limited e-waste management infrastructure. As noted by the monitor, if the rate of e-waste collection, management, and disassembly is brought to 60 per cent by 2030, the benefits would trump the costs while reducing continued human exposure [1].

For Oceania, per capita e-waste generation and collection and recycling stood at 16.1 kg and

6.7 kg respectively. It is also important to note that 8.2 percent of the total e-waste generated was shipped across borders from high-income to low- and middle-income countries, justifying how efficient management of e-waste through higher levels of automation must be scaled uniformly across national borders [1].

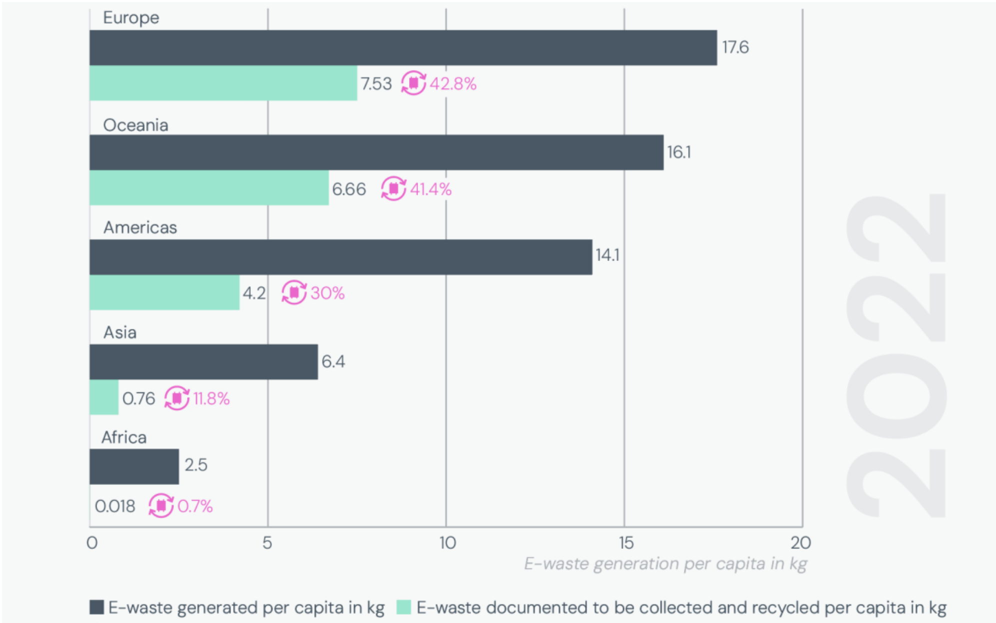


Figure 1.1: Global E-Waste Monitor. [1]

Australia specifically generated 511,000 tonnes of E-Waste, with the national total set to increase by 30 percent taking it to 657,000 tonnes by 2030. According to the Australian Department of Climate Change, Energy, Water, the Environment and Water, to manage the magnitude of waste generated, increased consumption, and imports, Australia has a product stewardship system to manage products and materials over their lifespan. This form of extended producer responsibility extends to management being voluntary, co-regulatory, or mandatory with no specific compulsory product stewardship scheme for e-waste disassembly and management. Taking into account legislation such as the National Waste Policy and the 2011 National Television and Computer Recycling Scheme (NTCRS), the extent of what is classified as e-waste under Australian legislation is limited. Involvement of relevant stakeholders, including consumers, large companies, retailers, and local governments, is also absent with no clearly defined roles. [4]

A study by Tasbirul Islam et al [5] delineates the categorisation of e-waste across six main categories of e-waste that can lead to environmental problems due to shorter product lifespans. These range from large temperature exchange equipment such as freezers, and air conditioners to small telecommunication equipment such as mobile phones. The same study underlines how currently most developed countries are habituated to export their e-waste to developing countries such as India or China and underdeveloped nations with limited Environmentally Sound Management System (ESM) with possible dire consequences in terms of socio-economic circumstances, public health, and environment of that place [6]. The focus of this thesis would be on small IT and telecommunication equipment, specifically hard drives usually shredded and discarded given their large security threats to cloud service providers. A study [7] underlines that commonality of hard drives and their appropriate size complementing robotic manipulation further justifying the emphasis of this study. A report by the BBC [8] pointed out how for large data centres managing hard drives is an issue of risk management, with 'clearing' the data as the least secure method to manage hard drives. Data deleted can be retrieved through specialised techniques making shredding, melting, incineration, and melting essential to protect data. In such a scenario, our thesis would explore the right mechanism to manage the disassembly of hard drives through the efficient detection of screws and in turn help concerned data companies manage vital information securely. As a prelude to the disassembly of hard drives, data can also be overwritten with new patterns for protection, using cryptographic erase, or ensuring drives have built-in encryption. Irrespective, disassembly and detection of screws in drives is integral for both environmental and security reasons. Thus, automated disassembly susceptible to external uncertainties [9] should be prioritised over manual and destructive disassembly requiring human exposure and environmental pollution. The fundamental motivations for this thesis are the management of e-waste to optimise resource utilisation, preserve embodied energy, strengthen automation against extraneous changes, and tap into the expanding market of e-waste management. Most importantly, non-automated destructive disassembly poses various ethical and environmental hazards across the developed and developing worlds, making computer vision-enabled algorithms essential to tackle these problems. The main focus of this thesis is to evolve a computer-vision algorithm capable of autonomously detecting screws in hard drives in an economically, environmentally, and socially sustainable manner focusing on enhancing Extended-Producer-Responsibility and avoiding massive data leaks. For this, our focus will be on computer vision models such as You-Only-Look-Once (YOLO) - a deep learning neural network meant for object detection (the what and where), U-Net for image segmentation and division into minute pixels and their identification as well as K-Means as a machine learning algorithm to cluster similar data together. These methods are supported by a comprehensive dataset creation, image classification and preprocessing. For this thesis, the efficiency across each model is further optimised and made better through assessment as per Precision, Recall, IoU, and the Dice Coefficient.

1.1 Aim

The main aim of this thesis is to autonomously detect screws from hard drives under dynamic conditions, thereby aiding the disassembly process, minimise human exposure and use of destructive methods while handling screw variations and environment uncertainties.

These variations could range across product structure, conditions, the volume of product inputs, quality, the extent of disassembly needed, and uncertainty of the outcome of disassembly operations. To develop this algorithm, our methodology focuses on primarily three computer-vision models: YOLO, U-Net, and K-means. As a prelude to implementing these approaches, our fundamental steps include evolving a thorough annotated dataset of hard drive images with visible screws and different light orientations. This will be followed by augmenting the dataset and evolving the computer-vision algorithm that ultimately facilitates identification of different screw orientations and placements while ensuring it is dynamic to real-life adjustments. The algorithm will further be optimised and potentially be made compatible with a robotic system made robust, with vast perception and operation capabilities, cost-effectiveness, scalability, and user-friendliness.

Scope of Thesis:

1. Study of prevailing computer vision techniques and e-waste management systems to understand shortcomings in disassembly, levels of automation, and system dynamism.

2. Evolving a computer vision algorithm built on image datasets capable of detecting and disassembling screws from a hard drive.

3. An extended goal would be to optimise the algorithm and assess it across various evaluation metrics such as Precision, IoU, and Recall.

1.2 Problem Statement

Traditional methods of managing e-waste and handling computer hard drives (HDDs), such as incineration and shredding, are environmentally hazardous and pose serious security risks due to the potential leakage of sensitive data. Hard drives, in particular, are critical to address due to their widespread use across computers and laptops, and their vulnerability to data and privacy threats. Detection of screws is a vital step in e-waste management: not only to ensure data security and promote environmental sustainability, but also because screws are prevalent in electronic equipment, both large and small. However, manual screw detection and disassembly is time-consuming, error-prone, and exposes workers to toxic substances. Currently, the absence of advanced, automated disassembly systems significantly limits secure data handling and material recovery. This is especially relevant in the Australian context, where electronic waste is projected to increase by 30 percent by 2030 (UN Global E-waste Monitor). Without innovation in disassembly processes, the opportunity for safe, efficient, and scalable e-waste management—particularly for hard drives—remains underutilized. In this context, the problem statement guiding this project is grounded in the need for a scalable, intelligent solution that addresses the inefficiencies and risks of current screw detection and disassembly methods. Our project focuses on developing an automated, computer vision-based algorithm using deep convolutional neural networks to detect and disassemble screws from hard drives. This approach aims to address the dual challenge of data security and environmental degradation while opening up possibilities for resource recovery, remanufacturing, and safer and more efficient electronic waste handling practices.

Within this problem context, the **research question** devised for this thesis is as follows: How can a deep learning-based computer vision algorithm be developed to automatically detect and assist in the disassembly of screws from hard drives to improve data security and enable efficient e-waste management?

Chapter 2

Literature Review

2.1 Introduction to AI and Its Subsets

A basic understanding of the differences between deep learning, machine learning, and neural networks is integral to understand the approaches deployed for this study. A report by IBM [10] underlines how machine learning, deep learning, and neural networks are all subsets where each encircles the next and resides within the overarching Artificial Intelligence system. AI has evolved to become a colloquial term; a supplementary existence that mimics human intelligence and cognitive functions. AI further divulges into Artificial Narrow Intelligence (chatbots and computer vision backed automated tasks), General Intelligence, and Super Intelligence (an ongoing field) marked by decreasing ability to complete a task. A step further, machine learning that allows for better optimisation such as businesses recommending products and services as per what the consumer has bought or interacted with before. Machine learning works best with labeled data sets under supervised learning conditions and can also process unstructured data, determining sets of distinguishable features - in our case the silver structure of small screws in hard drives. K-Means is an example of a machine learning method that can categorise e-waste on the basis of type, material, or hazard level. Deep Learning is another subset within machine learning where the former is responsible for most of the feature extraction process, resulting in limited required human intervention. Deep Learning, also called 'scalable machine learning' assists any algorithm in observing specific patterns that in turn cluster similar inputs properly. Screws of a specific orientation, colour, size, or structure can be grouped into specific categories based on identified similarities and features. Clearly, a deep-learning model (such as YOLO and U-Net) requires more data points to improve accuracy. Deep Learning algorithms are further backed by neural networks that replicate the way brain neurons operate. A neural network with three or more layers becomes a 'deep'-learning algorithm. YOLO and U-Net both are backed by convolutional neural networks that aid in identification, precise localisation whereas K-Means is an unsupervised machine-learning algorithm best for helping automated sorting systems.

Since YOLO, U-Net and K-Means are all Computer-Vision based models, it becomes integral to gain an understanding of what exactly computer-vision entails. Computer Vision as an emerging field of academic research plays a vital role in the e-waste management industry. As a part of artificial intelligence and deep learning, computer vision is used to perceive, detect, process, and analyse visual data like images and videos. This collection of unstructured information with specific colours and texture further provides insights to questions like 'Where are the screws located' or 'How many screws does this specific hard drive have' [11]. The answers to these questions can be fed into a comprehensive data system embedded in a robotic machine to accurately detect screws, manage data to avoid security leaks and avoid manual shredding of hard drives. After years of development across Optical Character Recognition, and Intelligent Character Recognition, it was only in 2017 that algorithms to sort waste were developed. The first step for any computer vision-based technology is the ability to consistently detect the device in question (here, hard drives). For this, it is essential the system must be fed with enormous amounts of related data to train it for recognition and detection of similar screws across different types of hard drives. In short, a tested and trained computer vision model has the ability to work in irregular environments and new, unfamiliar scenarios. Given the expanse of the e-waste problem, the growing use and potential of deep learning technologies, and the need to manage data found in hard drives efficiently, the thesis strives to incorporate these three problems into a dynamic computer-vision backed algorithm.

2.2 Computer Vision in E-waste Management

Research around the specific combination of detection of screws in a hard drive using computer vision is still limited. A study [12]around computer vision algorithms for inspection of external screw threads underlines the widespread use of screws and threaded connections accounting for about

15 percent of the total mechanical parts in machinery and equipment. Conventional methods to measure and inspect screws include coordinate measuring machines, profile projectors, screw thread micrometres, and laser measurements. Additionally, screws are highly complex exceeding thirty different geometrical and dimensional characteristics making these conventional methods time-consuming and expensive. The study also aptly notes how there's a need for a single detection and measurement system for all types of screw features with reduced detection time with no necessary human contact. This study aimed to develop a computer vision application with the ability to take pictures and determine aspects such as screw thread pitch, width, and specifications such as metric or imperial. The focus was on identification and improving worker productivity. Another study has also noted how bolted connections are not only commonly found due to their usage as connectors but also easier to detect as screws are highly standardised and overtly visible [13]. Within the field of machine learning, there is also a distinction between one-stage or two-stage object detectors in terms of their abilities to determine location coordinates with one-stage detection having lower accuracy but higher speed, represented in YOLO (You Only Look Once). Another study [12] on using computer vision technology to aid with construction waste recycling uses a Complete Coverage Path Planning (CCPP) algorithm specifically meant to equip a robotic system to autonomously cover every inch of an area. Through a path that seeps into every corner of the target area, CCPP seeks to avoid obstacles and ensures efficiency.

2.3 Existing Approaches in E-Waste Management and Computer Vision

A simplistic overview of a computer vision system used for screw detection and disassembly of electronic waste is given in the seminal research given by Foo et Al [9] wherein the system comprises a DSLR camera mount. ted to face LCD monitors. The camera is attached to a computer equipped with a screw detection program written in Python. Across four different steps from pre-processing, to detection through deep learning, visual reasoning and tailoring the model to fit external requirements are made compatible with the computer algorithm-something which we will focus on in our thesis as well specifically for hard drives. Similarly, Apple through Liam [14] has focused on the pre-processing step to categorise materials into homogenous streams. Liam, which is able to disassemble 1.2 million iPhone units per year, the goal is to 'de-manufacture' through automation rather than simply shredding and mixing all material together. Hard drives with their complex anatomy and security concerns make manual shredding ineffective as it becomes harder to segregate specific materials, retrieve data, or restore embodied energy and resources. Apple's Daisy is a step ahead of Liam [15] in terms of being more compact, reducing time to disassemble, and handling diverse device types. Daisy allows for a worker to feed a bin of iPhones into its chutes with the robot placing them on a conveyor belt for scanning and identification. Thereafter, each phone placed within a metal bracket is separated from its display before being sent to a cooling chamber to be frozen.

Figure 2.1: Liam: An Innovation Story. [2]

Figure 2.2: Sorted components post-disassembly using the Liam system. [2]

Tesla through Tesla Vision also employs computer-vision as a part of its autopilot and Full-Self-

Driving systems aiding in features like steering, lane-keeping, braking, and accelerating. This is done by training copious amounts of real-life driving data, in turn used to train neural networks (HydraNet architecture) that allows for the car to perform multiple tasks simultaneously [16].

Hard-drives posit a different, industrial-grade challenge that is different from relatively larger operations such as phone disassembly or self-driven cars. Due to their small components and negligible space for manipulation, making the recognition and taxonomy of screws more difficult. Due to their enormous potential for automation given the repeated nature of the task, disassembly of hard drives is a viable area for the application of deep neural networks as well as make the entire process generalisable. A study [7] by the Department of Computer Science at the University of Innsbruck delves into a visual intelligence scheme for hard drive disassembly across recycling routines. The study underlines how disassembly processes hold high-automation possibilities but present difficulties in adaptability as most End-of-Life products can be of varied nature or perhaps, damaged. A lot of devices within the same cluster also have 'intra-class variance' depending upon their brand and model - in this case, across differently sized and coloured screws. Robotic disassembly thus, can not just simply rely on an open-loop procedure, making product analysis an important aspect of finding the best disassembly strategy. The problem at hand is the lack of generalisable and environment-independent procedures to incorporate within the disassembly process with the application of Deep Convolutional Neural Networks (DCNN) as the plausible solution. Intra structural variance can be resolved by training and retraining on a large dataset (for screws in hard drives) and extended to all devices that use such parts (recognition of screws in any possible device). The study further evolves and defines a specific taxonomy of hard-drives in association with an industrial partner wherein two datasets each for parts and screws are created. Further, 80 % of the dataset is deployed as a training set on selected computer vision methods such as MobileNetV1, Cascade-RCNN, Mask-RCNN, and Cascade-MobileNetV. Screw detection took place through bifurcation of all inputs into screws or artifacts where the image is processed through the Hough Circle Transform technique used for detecting the edges of perfect or nearly perfect round edges. For the purpose of this thesis, we've adopted a more multi-pronged approach that implements and compares three different models that allow for feature separation, pixel-level segmentation, and real-time object detection (through bounding boxes). Apart from clustering, segmentation, and detection, the use of bounding boxes via YOLO allows for more robust detection across different types of hard drives, orientations, and screw types - something which is limited with the use of Hough Circles.

2.4 Current E-Waste Management Practices in Australia

Purchasing about five million computers every year, e-waste consumption in Australia is amongst the highest in the world [17]. To curb this potential problem, the National Television and Recycling Scheme covers products ranging from TVs to laptops and hard drives and mandates every exporting and importing company to pay for the end-of-life recycling for these products. Within Australia, there are numerous organisations that aid companies and businesses with e-waste recycling. Clean-away for instance focuses on ensuring the physical destruction of the actual hard-drive as a way to protect sensitive data [18]. Securis, an organisation specialising in recycling and destruction underlines physical destruction of hard-drives also comprises various methods such as degaussing, shredding, and micro-shredding as a way to remove all possible data threats. It is also important to note how the NTRS is primarily focused on households (with over 98 percent of the population having access to free-recycling services)and small businesses. For larger companies and corporates, e-waste management takes place through their specific hardware supplier or a third-party recycling company [19] with a hyper-focus on a non-cavalier approach for handling hard-drives. Most companies, handling hard drives means destruction and degaussing given the sensitivity of the data at hand. Destructive methods however are environmentally less-sound and make valuable materials such as aluminium, rare earth magnets, copper vulnerable to loss. Instead of secure and smart disassembly, current methods focus primarily on destruction.

2.5 Research Gap

From the literature review, it is evident that most research around computer vision is restricted to general disassembly tasks without an emphasised focus on automated detection and disassembly of intricate components like screws in HDDs. As opposed to a single-method approach, this thesis delves into three different models, comparing them and their transferability toward industrial-grade hard drives and not just large-scale electronics. Most importantly, given the data threats associated with the handling of hard drives, there's been limited exploration of an alternative to destructive shredding and degaussing. This thesis attempts to present a non-destructive, computer-vision guided alternative that would enable recovery of valuable materials, maintain data security, and prevent environmental compromise.

Chapter 3

Methodology Overview

For the first step for this thesis' methodological approach we captured and compiled more than 1000 images of screws in different lighting conditions and orientations so as to train our models on as natural and dynamic a dataset as possible. Further, we adopted a multi-pronged approach, training, testing, and optimising three different architectures which were YOLO, U-Net, and K-Means.

You-Only-Look Once or YOLO is essentially a deep-learning model that allows for real-time object detection through predicting bounding boxes across different objects and class probabilities in a single evaluation (simultaneous prediction of where, what, and how sure the model is about what it detects). As one of the first state-of-the-art deep learning models developed back in 2015 b6 Joseph Redomond [20], this thesis adopted YOLO given its accurate speed, robustness, and ability to aptly detect screws across varied orientations.

While there's been negligible application of **U-Net Image Segmentation** specifically for disassembly of e-waste, this computer vision subset allows for segmenting an image into multiple minute areas, magnifying and assigning a label to each pixel of the image. The dataset for U-Net has been created using the **Segment Anything Model** where precise masks of screw regions were generated that aided robust model training.

As an unsupervised algorithm, **K-Means** allows categorisation of a given dataset into different non-overlapping groups on the basis of feature similarity. **K-means**' primary objective is to minimise intra-cluster variance i.e ensure specific objects within a cluster are as similar to each other as possible and as distinct from other groups (maximising inter-cluster separation). It also helps with facilitating pattern recognition and spatial reasoning for disassembly tasks. We specifically chose K-Means given its lightweight, computationally inexpensive baseline method to cluster different screw features without the inherent complexity of deep-learning models.

To compare these three architectures and enhance our algorithm's performance and robustness, we've used several evaluation metrics. **Precision** reflects the model's ability to avoid any kind of false positives i.e tells us how many of the screws the model predicted were actually right. **Recall** on the other hand, assess the proportion of actual screw instances or how many real screws were detected and found. **F1 Score / Dice Coefficient** gives us a balance or harmonic mean between Precision and Recall, helping assess overall accuracy. Finally, Intersection over Union checks how much the predicted box matches the real screw's location i.e the overlap between the predicted and ground truth bounding boxes, serving as our spatial accuracy metric.

Chapter 4

Dataset Creation

For our thesis, the first objective was to create a high-quality dataset forming the backbone of the study given that the performance, reliability, and real-world adaptability of deep learning models depend heavily on the richness and variety of their training data. For this thesis specifically with an aim of enabling accurate screw detection in hard drives as part of a robotic e-waste disassembly system, it was essential to create a dataset that could handle variations in screw placement, lighting, and device orientation. To build such a dataset, a three-phase approach was followed:

1. Start with an existing screw detection dataset available on Roboflow which was executed through course of Thesis B
2. Expand it with custom-captured images under diverse real-world conditions as per Thesis B
3. Apply extensive data augmentation to further enrich the dataset and simulate practical challenges done during Thesis C

This chapter hopes to outline each of these steps in detail and explains how different augmentation techniques contributed to building a more robust training set.

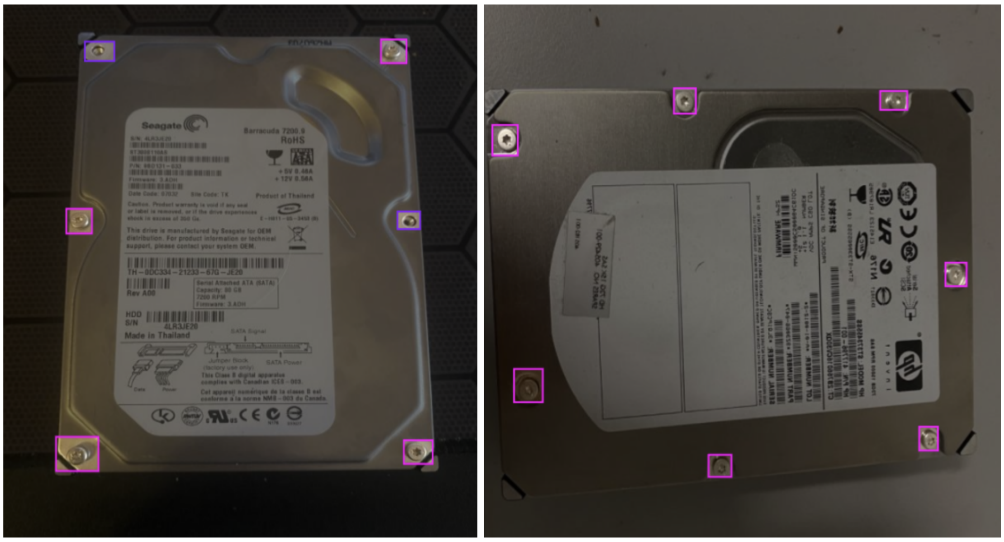


Figure 4.1: Initial bounding boxes around pre-existing dataset.

4.1 Dataset Collection Process

4.1.1 Baseline Dataset Acquisition

The initial foundation came from a publicly available dataset on Roboflow [21], created by the University of Limerick. This dataset contained images of hard drives with annotated screw locations and served as a solid starting point. However, it lacked diversity as most images were captured in consistent lighting, from top-down angles, and with similar types of screws. While useful for bootstrapping the model, it was clear that more variety would be needed for real-world adaptability and deployment.

4.1.2 Custom Image Collection

To bridge this gap, over 150 original images were collected using various hard drives. Care was taken to introduce realistic variability that the model might encounter in practical scenarios. This included:

- Lighting variations: Images were captured in bright, dim, and coloured lighting conditions to evaluate how well the model would perform under different exposures.

- Angle and orientation changes: Hard drives were photographed from multiple perspectives, including tilted and slightly rotated views, to encourage the model to learn orientation and how to perceive invariant features.
- Background and surface diversity: Drives were placed on different backgrounds — ranging from dark surfaces to wooden and white surfaces — to ensure that the model would not overfit to a particular texture or setting.

Post expanding our dataset, all collected images were manually annotated, where the bounding box coordinates were normalised and adjusted relative to the image dimensions. This ensured compatibility with the training pipeline and consistency across the dataset.

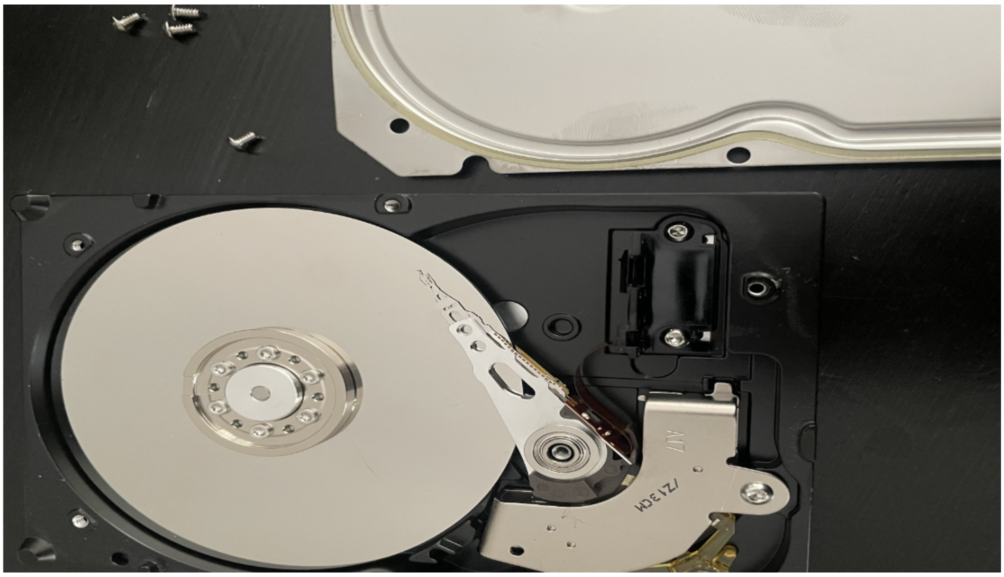


Figure 4.2: Real image added to the dataset.

4.2 Data Augmentation

Once the custom dataset was in place, the next step was to increase its diversity using data augmentation. This technique simulates different real-world conditions by applying a series of controlled distortions and transformations to existing images to further ensure our model is generalisable and operational in real-life environmentally intense settings as well. The objective was to expose the model to as many visual variations as possible — without having to collect thousands of real images. The Albumentations library was used to perform augmentation, as it provides a high-performance and flexible framework for image transformations. Below we've discussed an overview of the augmentation techniques used, along with their purpose.

Figure 4.3: Normal Image.

Figure 4.4: Augmented Image.

4.3 Techniques Used for Augmentation

```
transform = A.Compose([
    A.HorizontalFlip(p=0.5),
    A.RandomBrightnessContrast(p=0.5),
    A.RandomGamma(p=0.5),
    A.HueSaturationValue(p=0.3),
    A.Blur(blur_limit=3, p=0.2),
    A.CLAHE(p=0.2),
    A.GaussNoise(p=0.2),
    A.Rotate(limit=15, p=0.3),
```

Figure 4.5: List of Augmentation Techniques.

HorizontalFlip (p = 0.5) Randomly flips the image horizontally with a 50% probability. This simulates mirrored views of the hard drive and helps the model learn orientation invariance.

RandomBrightnessContrast (p = 0.5) Randomly adjusts the brightness and contrast of the image. This helps the model deal with inconsistencies in lighting conditions, such as underexposed or overexposed environments.

RandomGamma (p = 0.5) Applies a random gamma correction to simulate varying sensor responses and shadows, helping the model adjust to lighting differences more effectively.

HueSaturationValue (p = 0.3) Modifies the image hue, saturation, and value. This makes the model more robust to coloured lighting, camera settings, or tinted environments (for example, yellow or blue lighting).

Blur (blurlimit = 3, p = 0.2) Applies a mild blur with a kernel size up to 3. This simulates slight motion blur or out-of-focus conditions, which are common in handheld or moving camera setups.

CLAHE (p = 0.2) — Contrast Limited Adaptive Histogram Equalisation Enhances contrast in local image regions. This helps in low-light environments where screw details may otherwise be lost.

GaussNoise (p = 0.2) Adds Gaussian noise to the image. This prepares the model for real-world scenarios where sensor noise or compression artefacts may distort visual features.

Rotate (limit = 15, p = 0.3) Randomly rotates the image by up to ±15 degrees. This helps the model learn rotation-invariant features, which is essential when the hard drive is not aligned perfectly.

4.4 Final Dataset Structure

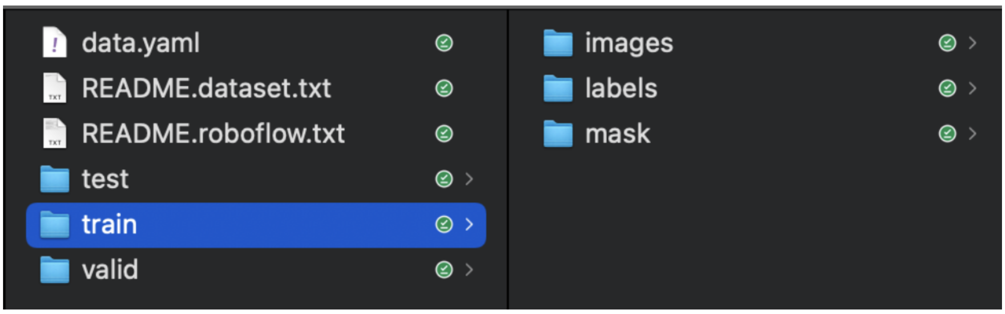


Figure 4.6: Final Data Structure.

After augmentation and cleanup, the dataset was split as follows:

- *Train folder:* 942 images, used for training the model.
- *Validation folder:* 46 images, used for monitoring validation performance during training.
- *Test folder:* 26 images, reserved for final evaluation of the trained model.

Each folders having Images, Labels in form of coordinates for the bounding boxes in form of txt files and Masks which were all the original images in segmented form that were later used for the Segmentation model.

The final dataset contained 1,014 images, offering a balanced mix of real-world complexity and simulated diversity. This ensured the model would not only learn to detect screws accurately but also generalise across challenging, unseen conditions that are often innate to the disassembly process.

(a) Front side of hard drive with screw annotations.

(b) Back side of hard drive with screw annotations.

(c) Data set expansion with new hard drive samples annotated for screw positions.

Figure 4.7: Screw position annotations on various hard drive surfaces used in the dataset.

Chapter 5

K-Means

K-Means clustering is a popular unsupervised learning algorithm that partitions data into k clusters based on feature similarity. In this project, K-Means was explored as a lightweight baseline for screw detection by leveraging spatial features extracted from hard drive images. Unlike supervised deep learning approaches, this method does not require extensive labelled data or complex model architectures, making it computationally inexpensive and interpretable.

5.1 Dataset Creation

Dataset and Annotations:

The dataset used for this experiment consisted of manually masked images of hard drives, including different segments for screws. We used annotation files to get the real coordinates or ground truth for verifying how good the model is. We found matching images, labels, and mask paths were programmatically retrieved using glob.

Preprocessing and Feature Extraction:

Each image was converted to grayscale and subjected to Gaussian blurring to reduce noise and enhance circular features. Circle detection was then performed using the **Hough Circle Transform**, which is effective at identifying round structures — a suitable heuristic for locating potential screws.

```python
# Apply Gaussian blur to reduce noise
blurred = cv2.GaussianBlur(gray, (9, 9), 2)
# Apply Hough Circle Transform to detect circles (possible screws)
circles = cv2.HoughCircles(blurred, cv2.HOUGH_GRADIENT, dp=1.2, minDist=50,
                           param1=100, param2=30, minRadius=10, maxRadius=30)
```

Listing 5.1: Preprocessing steps — Gaussian blur and Hough Circle Transform to detect circular features.

The output from this step was a list of (x, y) coordinates representing the centres of detected circular objects in the image.

Clustering with K-Means:

To classify detected circular features as either screws or non-screws (e.g., holes), K-Means clustering was applied with k = 2. The clustering was performed on the feature space formed by the (x, y) coordinates of the detected circles.

Post-clustering, each detected object was labelled as either a screw or a non-screw based on its cluster assignment. Visual feedback was generated by colour-coding the clustered objects (e.g., green for screws and red for non-screws) using OpenCV.

5.2 Results

Model performance was quantitatively evaluated using **Precision** and **Recall** by comparing the predicted circle coordinates with the annotated ground truth. A projected coordinate was considered a true positive if it was within a Euclidean distance threshold (15 pixels) from a ground truth screw.

```python
precision = tp / (pred_len + 0.001)
recall = tp / (gt_len + 0.001)
```

Listing 5.2: Precision and recall calculation formulas used to evaluate the clustering performance.

The final scores were averaged across the dataset, providing an objective view of the clustering model's accuracy.

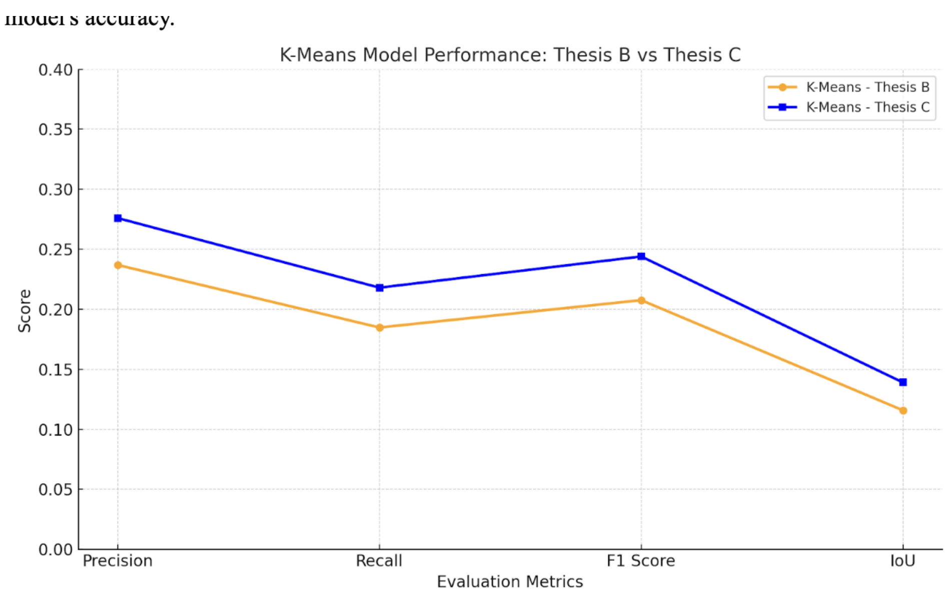


Figure 5.1: Results Comparison for K-Means between Thesis B and Thesis C as per Evaluation Metrics.

As shown in the graph, the K-Means approach offered only modest improvements between Thesis B and Thesis C. The **relatively low recall and IoU values highlight the limitations of relying solely on geometric heuristics** and clustering without contextual understanding.

However, K-Means was valuable as a fast, explainable baseline that established performance bounds without the need for GPU resources or labelled training.

5.2.1 Summary

The performance of the K-Means clustering approach was evaluated using precision and recall by comparing predicted screw coordinates to annotated ground truth, with a 15-pixel Euclidean distance threshold defining true positives. While this method provided quick and explainable results without requiring labeled data or GPU resources, its performance remained at best modest. The final scores, averaged across the dataset, indicated that K-Means offered only slight improvements from Thesis B to Thesis C. The approach's reliance on simple geometric features limited its ability to accurately distinguish screws, particularly in complex scenes with visually similar components.

5.2.2 Conclusion

Although the K-Means clustering method was limited in terms of segmentation accuracy, it served as an effective baseline to benchmark the performance of more advanced models. Its lightweight, unsupervised nature made it suitable for initial experimentation and performance bounding. However, the lower recall and IOU values underscore the limitations of approaches based solely on simple shape-based rules and spatial patterns. This reinforces the need for learning-based, data-driven methods that can be understood from context and visual patterns, particularly in dense environments such as screw detection.

Chapter 6

U-NET

U-Net is a widely used deep learning architecture that was first introduced in the "U-Net: Convolutional Networks for Biomedical Image Segmentation" paper (20). The primary purpose of this architecture was to address the challenge of limited annotated data in the medical field by focusing on data augmentation of existing annotated samples and using them more efficiently. This network was designed to effectively leverage a smaller amount of data while maintaining speed and accuracy.

6.1 Architecture

The architecture of U-Net is unique in that it consists of a contracting path and an expansive path. The contracting path, which consists of encoder layers that help capture contextual information, reduces spatial resolution of the input. It acts like a tunnel where the image gets smaller to allow for more relevant information to be retrieved. The expansive path on the other hand contains decoder layers, where the encoded input is enlarged again and a segmentation map is produced that connects all the relevant patterns learned through the contracting path [22].

The contracting path in U-Net is responsible for identifying the relevant features in the input image. The encoder layers perform convolutional and pooling operations that reduce the spatial resolution of the feature maps while increasing their depth, thereby capturing increasingly abstract representations of the input.

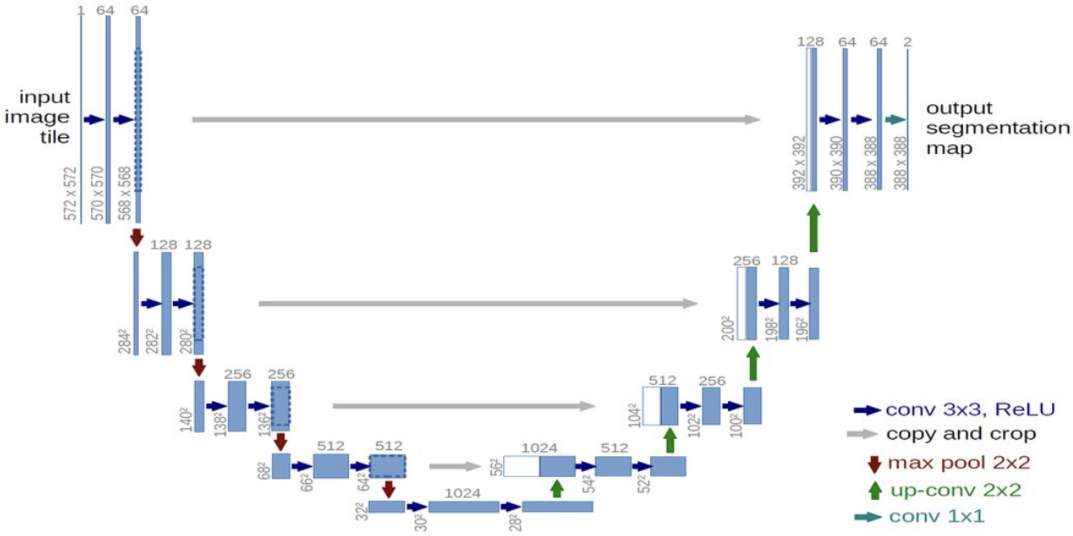


Figure 6.1: U-Net architecture diagram illustrating the encoder-decoder structure with skip connections for precise image segmentation.

This contracting path is similar to the feedforward layers in other convolutional neural networks. On the other hand, the expansive path works on decoding the encoded data and locating the features while maintaining the spatial resolution of the input. The decoder layers in the expansive path upsample the feature maps, while also performing convolutional operations. The skip connections from the contracting path help to preserve the spatial information lost in the contracting path when the image was made smaller, helping the decoder layers to locate the features more accurately once the image is made large again.

A 1*1 convolution follows the last decoder block with sigmoid activation which created the final segmentation mask where each pixel gets classified. This way, it could be said that the contracting path passes across information to the expansive path thus, we can capture both the feature information and localisation with the help of a U-Net - understanding both what's in the image i.e features and where everything is i.e localisation.

6.2 Dataset Creation

Dataset creation in the image segmentation task is challenging as we have to provide the pixel mask of each object/class we will identify. In our case we are going to identify screws and holes, so we need to create a mask image depicting the location of the screws and holes within the image. Masking every pixel is challenging, as it requires the right tools and detailed pixel-level labels, which can be computationally expensive for high-resolution images. Another difficulty arises with objects that have ambiguous edges, like screws in our case, leading to inconsistent annotations. Mislabeled pixels can significantly degrade model performance, making rigorous review processes essential.

6.2.1 Overcoming Dataset-related Challenges

To mitigate this challenge, we took the aid of Semi-Automated Tools: such as the Segment Anything Model to speed up annotation. SAM can generate masks based on various input prompts like Points (e.g., user clicks on an object) , Bounding boxes (rough region of interest), Text descriptions (limited integration, but possible with extensions).

We used SAM to leverage these capabilities to automate the process of image segmentation which in our case was providing the bounding box points of the screws and holes . SAM automatically creates a segmentation mask of these objects and thus creates a Segmentation mask image , which we use to give it to our UNET model.

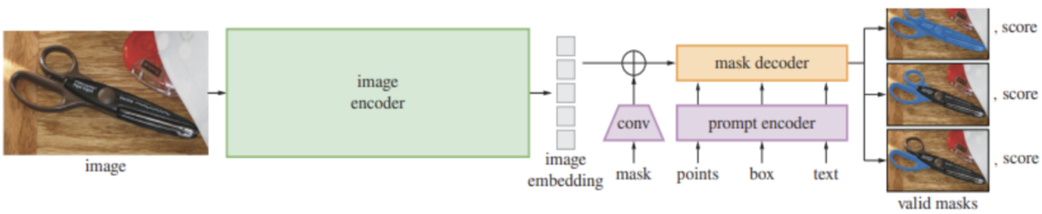


Figure 6.2: Architecture of the Segment Anything Model (SAM) which allows generation of multiple object masks through an input image.

SAM uses Zero-shot generalization as it has learned a general idea of what objects look like. As a result, SAM is able to recognise newer objects never seen before without the need for more training i.e it acquires zero-shot generalisation. SAM's advanced capabilities are the result of its training on millions of images and masks collected through the use of a model-in-the-loop data engine. Researchers used SAM and its data to interactively annotate images and update the model. This cycle was repeated many times over to improve both the model and the dataset.

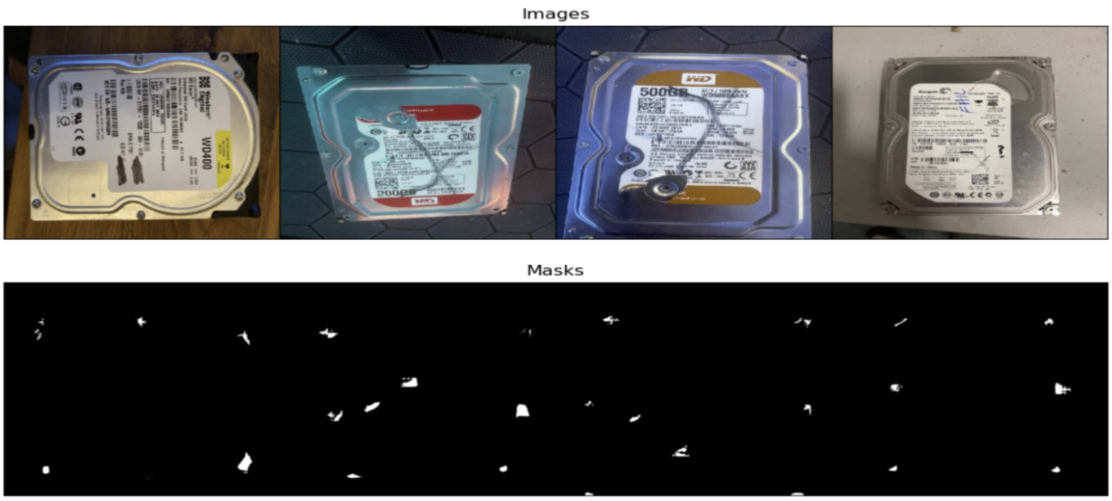


Figure 6.3: Images and their Masks within SAM model.

We used the SAM library to generate the model registry and added GPU as device to run SAM on GPU as the model is highly compute intensive . Providing the image and Datapoints , SAM created a masking image of each object we need to identify in the dataset.

```
sam_checkpoint = "sam_vit_h_4b8939.pth"
model_type = "vit_h"
sam = sam_model_registry[model_type](checkpoint=sam_checkpoint)
sam.to(device=device)
mask_generator = SamAutomaticMaskGenerator(sam)
predictor = SamPredictor(sam)

masks, scores, logits = mask_predictor.predict_torch(
    point_coords = None,
    point_labels = None,
    boxes = transformed_boxes,
    multimask_output = False
)
```

```
all_masks = masks.detach().cpu().numpy()
```

Listing 6.1: Code for loading the SAM model checkpoint and generating segmentation masks using bounding boxes.

6.3 Results

```
842/842 - 197s - 234ms/step - accuracy: 0.9949 - dice_coef: 0.6310 - iou_metric: 0.5097 - loss: 0.0052 - val_accuracy: 0.9941 - val_dice_coef: 0.3707 -
val_iou_metric: 0.3085 - val_loss: 0.0206
Epoch 12/50
842/842 - 197s - 234ms/step - accuracy: 0.9951 - dice_coef: 0.6602 - iou_metric: 0.5401 - loss: 0.0047 - val_accuracy: 0.9941 - val_dice_coef: 0.3643 -
val_iou_metric: 0.3040 - val_loss: 0.0199
Epoch 13/50
842/842 - 197s - 234ms/step - accuracy: 0.9952 - dice_coef: 0.6803 - iou_metric: 0.5611 - loss: 0.0045 - val_accuracy: 0.9940 - val_dice_coef: 0.3663 -
val_iou_metric: 0.3061 - val_loss: 0.0227
...
Epoch 20/50
842/842 - 197s - 234ms/step - accuracy: 0.9956 - dice_coef: 0.7872 - iou_metric: 0.6853 - loss: 0.0030 - val_accuracy: 0.9941 - val_dice_coef: 0.3769 -
val_iou_metric: 0.3164 - val_loss: 0.0241
Epoch 21/50
842/842 - 197s - 234ms/step - accuracy: 0.9956 - dice_coef: 0.7944 - iou_metric: 0.6943 - loss: 0.0029 - val_accuracy: 0.9941 - val_dice_coef: 0.3712 -
val_iou_metric: 0.3134 - val_loss: 0.0250
```

Figure 6.4: Training log showing accuracy, Dice coefficient, IOU metric, and loss over epochs.

The U-Net model demonstrated a steady and consistent improvement in segmentation performance across training epochs, as indicated by the upward trends in the Intersection over Union (IOU) metric across different experimental runs. In the early stages of training (first 20 epochs), the IOU score shows a gradual increase, reflecting the model's initial learning phase where it begins to capture the basic structure of the target objects. As training progresses to 60 and eventually 100 epochs, the model continues to refine its predictions, with the IOU metric rising steadily and showing fewer fluctuations. The final IOU values reach significantly higher levels compared to the initial epochs, indicating successful convergence. This progressive increase in IOU highlights U-Net's strong ability to learn pixel-level details and improve segmentation accuracy over time, making it a highly effective choice for tasks requiring precise object localisation and boundary delineation, even in complex or noisy datasets.

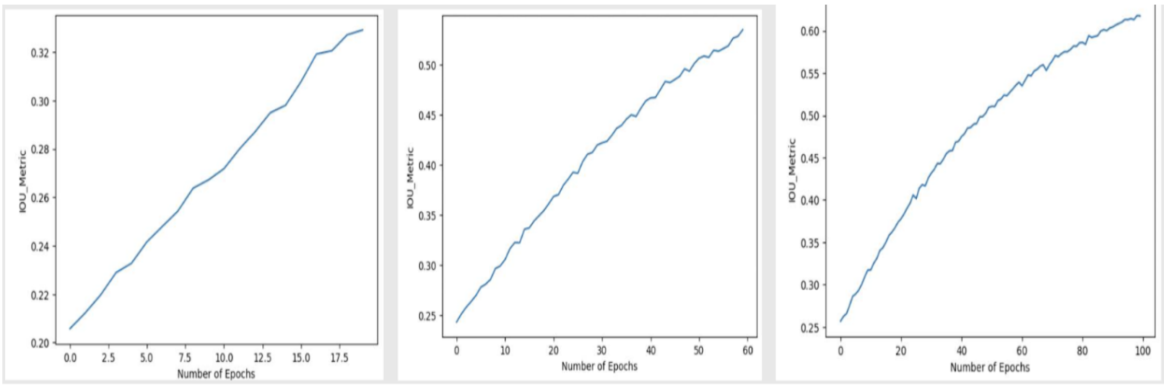
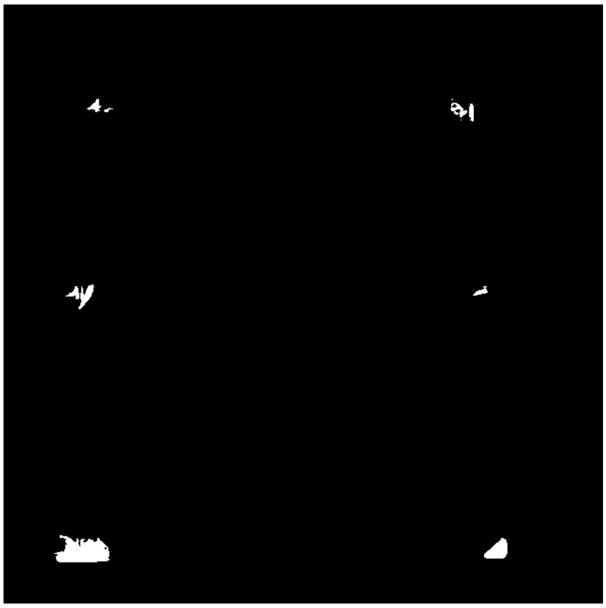


Figure 6.5: U-Net Results using IoU Metric.

(a) Testing Image.



(b) U-Net Results.

Figure 6.6: Comparison of segmentation results from U-Net model with the original testing image.

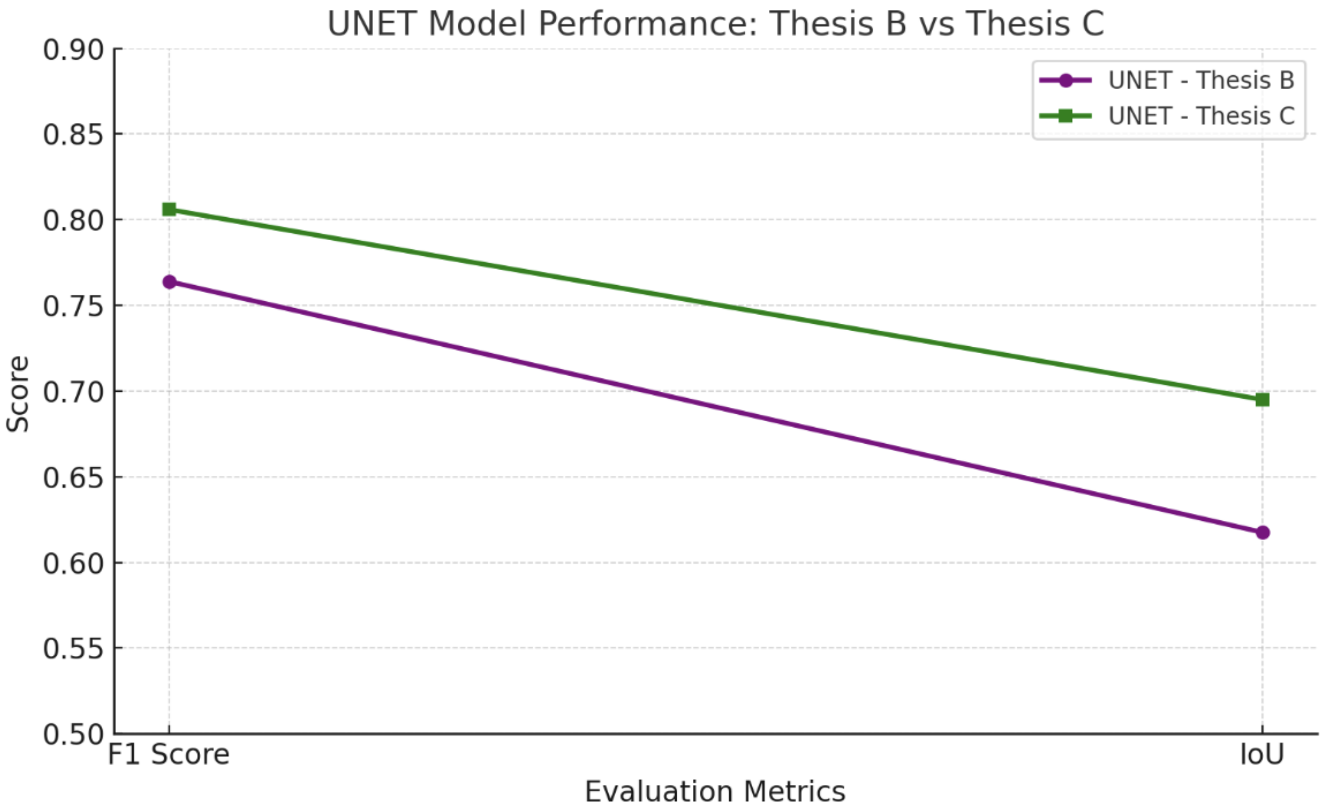


Figure 6.7: Results Comparison for U-Net between Thesis B and Thesis C as per Evaluation Metrics.

To further check the optimisation between thesis C and thesis B, the IOU and F1 score was calculated on both the values and there was a significant improvement in both due to the data augmentation done to the dataset. As the model was trained on a much larger data with varied brightness, colours, and rotations **it was easier for the Unet model to detect screws in Thesis C than Thesis B as the IOU increased from 0.61 to 0.69 and F1 score increased from 0.76 to 0.80**. **F1 score denotes a smart average of Recall and Precision as it balances both. Since U-Net is a segmentation model, we can only estimate the values of precision and recall from the F1 Score to understand how efficiently the two work together.**

6.3.1 Summary

As seen, the U-Net model exhibited consistent improvements in segmentation performance throughout training - evident from a steadily increasing IOU score. Through the initial 20 epoches, the model learned the basic structure of different target objects which was further refined through 60 to 100 epochs leading to more stable IOU values. When comparing results from Thesis B and Thesis C, the application of data augmentation techniques in Thesis C, such as adjustments in brightness, color, and rotation resulted in a notable boost in performance. The IOU improved from 0.61 to 0.69, and the F1 score rose from 0.76 to 0.80, demonstrating enhanced model generalisation and robustness in screw detection.

6.3.2 Conclusion

Our results conclude that U-Net is a powerful segmentation model capable of learning fine-grained pixel-level features and improving performance over time. Moreover, the performance gains observed in Thesis C highlight the importance of data augmentation in training deep learning models for real-world tasks. By diversifying the training data, the model becomes more adaptable and accurate, particularly in complex environments such as e-waste disassembly. Through this, we can reiterate the suitability of U-Net for precise object location, making it a strong architecture for automated visual disassembly and detection.

Chapter 7

You Only Look Once - YOLO

In this chapter, we present the implementation of object detection using the YOLO (You Only Look Once) framework. YOLO is a widely adopted real-time object detection system, celebrated for its exceptional speed and accuracy. YOLOv8, which has been used for this study, introduces a thoughtfully refined architecture that builds on the strengths of previous YOLO versions while making key improvements in speed, accuracy, and versatility.

7.1 Architecture

Its architecture is divided into three main parts: the Backbone, the Neck, and the Head.

1. **The Backbone** is responsible for extracting deep, meaningful features from the input images. It uses an enhanced CSP (Cross Stage Partial) design along with C2f modules, which allow the model to retain important information while keeping the network lightweight and computationally efficient.

2. Once features are extracted, they are passed to **the Neck**, which plays a crucial role in combining and refining these features across different scales. The Neck is designed using a combination of Feature Pyramid Networks (FPN) and Path Aggregation Networks (PAN), ensuring that the model can detect objects of all sizes - from the smallest screws to larger structures both with equal effectiveness.

3. One of the biggest changes in YOLOv8 lies in **its Head**. Unlike older YOLO versions that coupled classification and bounding box regression into a single branch, YOLOv8 decouples them into separate pathways. This simple yet powerful change allows the model to specialise better in each task, leading to faster convergence during training and higher accuracy during inference [23].

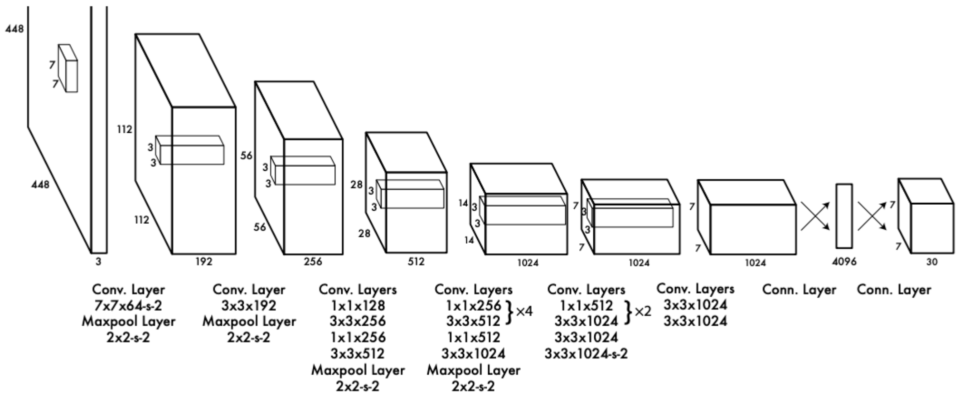


Figure 7.1: The YOLO Architecture. [3]

7.2 Dataset Creation

The dataset used for training was prepared and annotated using Roboflow, a platform that helped streamline the dataset curation process. The dataset was exported in YOLOv8-compatible format with the corresponding data.YAML file specifying class names and data paths. It included variations in screw size, orientation, and lighting that helped replicate real-world disassembly scenarios.

The model training was executed in two progressive stages, adopting a sequential fine-tuning approach within the same domain. In the initial phase (Thesis B), a YOLOv8n architecture was trained from scratch (i.e., with randomly initialised weights) using a dataset of 500 annotated images. This base training stage enabled the model to learn fundamental visual representations and spatial patterns associated with screw detection. In the subsequent phase (Thesis C), the dataset was expanded to 1,000 images through a combination of additional data collection and image augmentation techniques (such as horizontal flipping, brightness adjustment, and scaling), thereby increasing both the volume and variability of the training data.

```
# Define training parameters and start training
```

```
results = model.train(
    data=dataset_yaml,          # Path to dataset YAML file
    epochs=50,                  # Number of epochs (adjust as needed)
    name='screw_detection'      # Experiment name (optional)
)
```

Listing 7.1: Code snippet to configure and initiate YOLO model training with dataset path, epoch count, and experiment name.

Instead of reinitialising the model, the previously trained weights were used as the starting point for continued training on the larger dataset. This method of intra-domain weight transfer allowed for the preservation of earlier feature learning while enabling further refinement as the model got used to a larger set of inputs. While not classified as traditional transfer learning from a general-purpose dataset, this domain-specific weight reuse acted as an internal knowledge transfer mechanism. It improved convergence stability and performance across diverse conditions, particularly in terms of generalisability to different orientations, backgrounds, and screw colours.

7.3 Results

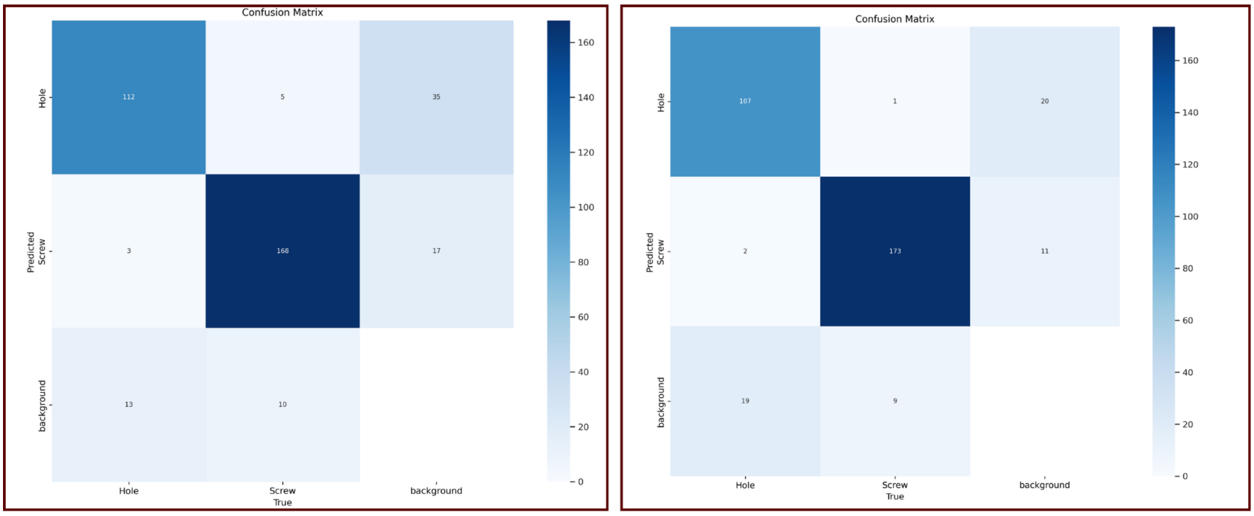


Figure 7.2: Confusion matrices illustrating the classification performance of the object detection model for Thesis B and Thesis C showing true positives, false positives, and false negatives across the classes: hole, screw, and background.

These were the confusion matrices generated by the algorithm for Thesis B and Thesis C. As we can clearly see the True positives for detecting screws have increased from 168 to 173, and the False positives and negatives have decreased as well on the same testing data showing clear signs of model optimisation for screw detection from Thesis B to Thesis C.

To further confirm this the YOLO model's performance was quantitatively assessed using standard object detection metrics, including **precision, recall, F1 score (Dice coefficient), and Intersection over Union (IoU)**. These metrics collectively measure the model's ability to accurately and consistently detect screws under various conditions.

To reiterate what we had mentioned in our methodology overview, **Precision** quantifies the proportion of true positive predictions among all predicted positives, reflecting the model's ability to avoid false positives. Similarly, **Recall** measures the proportion of actual screw instances that were correctly identified, indicating the model's effectiveness in minimising false negatives. Further, **F1 Score / Dice Coefficient** provides a harmonic mean of precision and recall, offering a balanced measure of accuracy. And finally, IoU assesses the overlap between the predicted and ground truth bounding boxes, serving as a spatial accuracy metric.

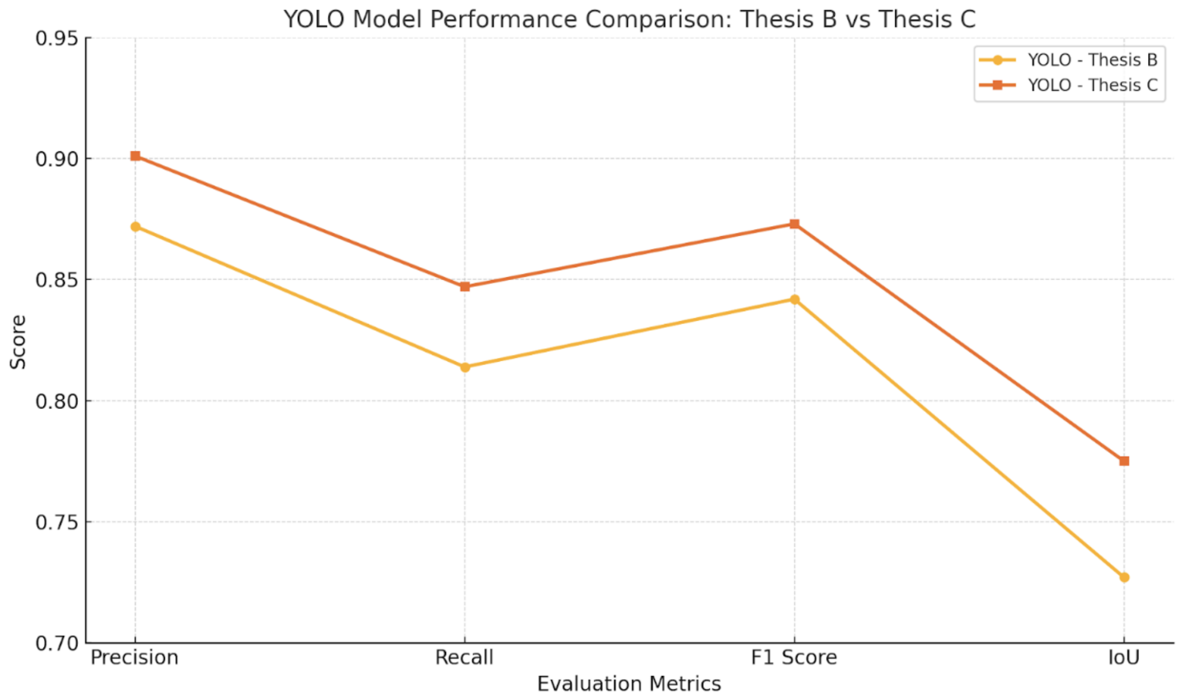


Figure 7.3: Results Comparison for YOLO between Thesis B and Thesis C as per Evaluation Metrics.

7.3.1 Summary

The YOLO model demonstrated a consistent improvement across all evaluation metrics from Thesis B to Thesis C. **Precision increased from 0.872 to 0.901**, indicating better suppression of false positives. **Recall also improved from 0.8139 to 0.847**, suggesting enhanced detection completeness. **The F1 score rose from 0.8419 to 0.873**, confirming a balanced gain in both precision and recall. Finally, **the IoU improved from 0.727 to 0.775**, highlighting more accurate localisation of screws in the images. These improvements validate the effectiveness of continued training on an expanded dataset and affirm YOLOv8's capability for high-performance detection in practical, domain-specific tasks.

7.3.2 Conclusion

From this, we can ascertain **that YOLOv8 proved to be an effective tool for this object detection task**. Its balance of speed, accuracy, and adaptability enabled robust screw detection in diverse scenarios. The trained model which has been aptly optimised can now be potentially integrated into a robotic disassembly pipeline to support automated e-waste processing and component recovery without unnecessary destruction.

Chapter 8

Results and Analyses

In order to understand and discern which architecture is best for our thesis objective, this chapter focuses on comparison of YOLO, U-Net, and K-Means as per different evaluation metrics. This would not only pinpoint the best model for practical effecitveness and performance but also tell us their relative strength, weaknesses, and limitations.

8.1 Compararing through different Evaluation Metrics

The models were evaluated based on standard performance metrics, namely Precision, Recall, F1 Score (Dice Coefficient), and Intersection over Union (IoU). These metrics were chosen to offer a comprehensive view of both detection accuracy and spatial localisation quality.

A combination of standard performance metrics namely Precision (helps avoid false positives i.e determine the rightness of the detection), Recall (how many screws were actually detected), Dice Coefficient (an average of Precision and Recall) and the Intersection Over Union (to determine location accuracy).

The table above presents a succinct comparison of the three architectures; KMeans, U-Net, and YOLO — highlighting their training times and corresponding performance metrics. Clearly, YOLO offers a balanced trade-off between training efficiency and model accuracy, achieving high Dice and IOU scores in significantly less time than U-Net. While K Means was the fastest to train, its

METRICS	THESIS B			THESIS C		
	K Means	UNET	YOLO	K Means	UNET	YOLO
Precision	0.2369	-	0.872	0.276	-	0.901
Recall	0.1848	-	0.8139	0.218	-	0.847
F_1 score / Dice Coefficient	0.2076	0.764	0.8419	0.244	0.806	0.873
IOU		0.1158	0.6175	0.139	0.695	0.775

Table 8.1: Comparison of K Means, UNET, and YOLO metrics across Thesis B and Thesis C.

	KMeans	Unet	YOLO
Time to Train	5 minutes	13 hours	4 hours
Dice	0.24	0.80	0.87
IOU	0.13	0.69	0.77

Table 8.2: Comparison for different devices as per Training Time.

performance lagged considerably behind the deep learning models. This reinforces the idea that although faster methods exist, more sophisticated architectures like YOLO can deliver superior results within a reasonable computational budget, making them more suitable than the rest.

Key observations:

- YOLO consistently outperformed K-Means and U-Net across all metrics.
- The improvements from Thesis B to Thesis C across all models validate the importance of expanding and augmenting the dataset.
- Point of comparison was Industrial Robotics Apple's Daisy [24]), Automated vehicles (e.g. Tesla)

8.2 K-Means

K-Means clustering offered a very lightweight and computationally efficient approach for initial segmentation tasks. One of its biggest advantages was that it did not require any labeled data, operating as a fully unsupervised machine learning method. This made it extremely quick to implement, especially in situations where manual annotation would have been time-consuming and resource-intensive. In real-world applications, such a method could be particularly valuable during very early-stage prototyping, where the goal is simply to explore rough patterns in the data rather than achieve production-level accuracy. An instance could be segregating metallic components from plastic casings when provided with a huge pile of e-waste without detecting and separated individual screws or precisely outlining where an object ends and the other begins. It can howevever serve as a useful baseline when datasets are small, incomplete, or entirely unavailable.

However, the segmentations produced by K-Means were often coarse, meaning that they lacked fine boundary details and tended to group nearby pixels together without fully respecting object edges. The method also introduced considerable noise i.e randomly mis-clustered pixels that did not correspond meaningfully to actual objects — particularly in complex scenes. These inaccuracies made K-Means unsuitable for precision-critical applications like robotic disassembly, where even minor errors in object boundaries could lead to operational failures. Moreover, K-Means struggled in cluttered environments with overlapping or irregularly shaped objects, as its reliance purely on pixel similarity (such as color or position) failed to capture more abstract or structural features. While its simplicity and speed made it attractive for preliminary experimentation, K-Means ultimately proved too unreliable for deployment in any setting requiring high segmentation fidelity.

8.3 U-Net

U-Net as a method is ideal for helping with effective pixel-perfect segmentation, making it exceedingly well-suited for identifying fine-grained details, such as small, overlapping, or partially hidden screws. Its architecture, with skip connections that help the model preserve integral details such as edges and shapes even as the image is processed across layers. This enabled it to achieve high spatial accuracy, allowing the model to precisely differentiate between closely located objects and capture even subtle features. In real-world scenarios, U-Net would be ideal for highly controlled environments where fine precision is critical, such as medical imaging, microelectronics inspection, or precise manufacturing tasks, where even minor segmentation errors could have significant consequences.

However, these advantages came at the cost of heavy computational demands. U-Net required large volumes of meticulously annotated, pixel-level training data, which is expensive and time-consuming to generate at scale. Additionally, the model exhibited longer training times and slower inference speeds compared to lighter object detection approaches. This created challenges for real-time applications such as robotic disassembly, where rapid and real-time decision-making is essential. Unless heavily optimised through techniques like model pruning, quantization, or using more lightweight variants of U-Net deploying the full model in real-time industrial settings could be impractical. Therefore, while U-Net offers remarkable segmentation quality, its resource intensity poses significant barriers for time-sensitive and cost-sensitive deployment environments.

8.4 YOLO

Through extensive comparisons, evolving datasets, and multiple rounds of optimisation, YOLO emerged as one of the strongest candidates for building our real-time detection algorithm. It consistently demonstrated high accuracy in identifying key components like screws, even when tested under different lighting conditions and across a variety of device surfaces. One of YOLO's standout advantages was its impressive speed where it achieved real-time detection with relatively low computational demands, making it highly suitable for industrial applications where responsiveness is critical. In real-world scenarios, YOLO performed particularly well for medium-sized screws, maintaining detection accuracy despite changes in colour, material, or minor variations between devices. However, it did come with some limitations. Compared to models like U-Net, YOLO was slightly less effective for tasks requiring pixel-perfect segmentation; while it could accurately detect the presence and general location of screws, it sometimes struggled to precisely trace their exact edges, especially when screws were small, overlapping, or partially obscured. Additionally, YOLO required object-level labels (bounding boxes) during training. While this labelling process was far less demanding than full pixel-wise annotation, it still involved careful manual work to ensure accuracy.

Another challenge was that YOLO's performance could drop when encountering screws that were either significantly smaller or larger than the examples seen during training, highlighting the model's sensitivity to size variations. This underlined the importance of including a wide range of object sizes and appearances during dataset preparation to ensure strong generalisation for real-world dynamic scenarios.

8.5 Testing on Different Devices

Figure 8.1: Screws detected on a projector and modem.

The YOLO model was further tested on devices beyond hard drives to evaluate its real-life application and transferability. The following outcomes were observed:

- Medium-sized screws on different devices such as projectors, modems, and stands were reliably detected, regardless of material colour or minor lighting differences.
- Screw colour did not affect detection; both dark screws and metallic shiny screws were successfully identified.
- Very small screws were occasionally missed, especially those with shallow imprint depths or unusual reflections.
- Very large screws (significantly bigger than training examples) were also sometimes not detected correctly, as the model was primarily trained on screws of moderate size.

8.5.1 Limitations

Figure 8.2: Screws detected on a Macbook Pro and Pedestal Fan.

While the overall performance of YOLO was strong, the system exhibited certain limitations:

- The model struggled with detecting screws that were extremely small, shallow, or highly reflective under intense lighting conditions.
- Very large screws, especially those that differ significantly from the size range present in the training data, were occasionally misclassified or missed.

- For precise disassembly tasks that require pixel-level se.g.mentation (for example, identifying screw edges versus screw heads), the YOLO bounding box approach was less effective compared to segmentation models such as U-Net.

- Variability in background textures and complex textured surfaces could occasionally introduce false positives, although this was relatively rare.

These limitations suggest that while YOLO is a strong candidate for fast and reliable screw detection, integrating it with more fine-grained post-processing techniques, or using it in combination with segmentation models, could further improve the system for industrial-scale robotic disassembly tasks.

Chapter 9

Conclusion

This thesis set out to address a specific and emphasised bottleneck in the field of e-waste disassembly processes: the automatic and reliable detection of screws in hard drives. Given their pervasive presence and the massive security challenge posited by hard drives, this thesis set out by recognising the environmental and socio-economic drawbacks of simplistic destruction and incineration. Focusing instead on scalable and robust disassembly, our methodology adopted a multi-pronged approach, training, testing, and comparing three distinct architectures to finally arrive at a computer-vision based framework capable of meeting our challenge. Through the results, YOLOv8 emerged as the most viable and promising model out of the three, offering a balance between detection accuracy, inference speed, and computational speed. Nevertheless, this multi-level comprehensive approach allowed us to specify the strengths and specific utilisations of each model such as U-Net for superior pixel-precision despite its heavy computational demands and slightly limited real-time applications or K Means' computational lightweightedness despite its decreased precision. Ultimately, YOLOv8 with a Dice Coefficient of 0.873 and IoU of 0.775, opened the scope for generalisation across different devices and environmental conditions, establishing it as the most viable choice for industrial scale integration. Nevertheless, the study also highlighted key limitations. The model struggled with extremely small or highly reflective screws and showed occasional misclassification when encountering screw sizes outside its training range. These findings underline the importance of continuous dataset expansion, model fine-tuning, and, potentially, hybrid solutions that combine detection and segmentation for critical precision tasks. The research lays a solid foundation for future work in several directions: deploying the model on physical robotic systems, expanding the training dataset to include a broader range of e-waste devices, and integrating multi-modal sensing (e.g., depth or infrared imaging) to enhance robustness under real-world conditions. Beyond its immediate technical contributions, this work has broader significance. With global e-waste volumes surging and manual disassembly posing safety, economic, and environmental risks, the development of autonomous, intelligent disassembly systems is crucial. By enabling faster, safer, and more efficient component recovery, this research supports the transition toward a circular economy, helping to conserve valuable resources, reduce environmental impact, and promote sustainable manufacturing practices. It's a straightforward alternative to over simplistic destructive methods, presenting itself as an essential pre-processing step to carefully remove data, retrieve materials, and then destroy in environmentally sound ways. In conclusion, this thesis demonstrates that deep learning and computer vision driven screw detection is not only feasible but also highly impactful in advancing automated e-waste recycling. It serves as a stepping stone toward the development of fully autonomous robotic systems capable of handling the growing global challenge of electronic waste, and paves the way for future innovations in sustainable technology and intelligent automation.

Chapter 10

Future Work

Hardware Integration and Real-World Deployment:

A key future direction involves moving beyond controlled testing environments and integrating the model into a fully working robotic system. So far, the detection models have been tested mostly in simulated or ideal controlled conditions. However, when deployed in the real world, new challenges are bound to appear such as sensor noise (random errors in camera or sensor readings), partial occlusions (when parts of a screw are hidden from view), mechanical inaccuracies (robots missing their exact targets due to small errors in movement), and unexpected hardware failures (like cameras, motors, or arms not working properly).

Testing in real-world conditions is important to see whether the model can still perform reliably despite these imperfections. Furthermore, the next big step is to connect the screw detection system directly to a robotic arm or disassembly machine. This would allow the system not just to detect screws, but also to physically interact with them through unscrewing parts, dismantling devices, and moving towards full automation. Achieving this would help transform the project from a research prototype into a solution that could be used in actual recycling plants or industrial setups.

Model Generalisation and Scalability:

Another important goal for the future is making the model more adaptable and scalable to different types of e-waste. Right now, the training and testing focused mainly on hard drives. However, real recycling environments deal with a wide variety of products such as smartphones, laptops, tablets, and many others, each with different layouts, screw types, materials, and wear conditions.

To improve the model's robustness, future versions should be trained on much larger and more diverse datasets that include different devices, lighting setups, backgrounds, and examples of wear and tear, like rust, scratches, or broken components. Including damaged, old, or heavily used devices is important because in real-world recycling, items are rarely in perfect condition. A model that can reliably detect screws even on worn-out, dirty, or broken devices would be much more practical and valuable for real deployment.

Multi-Modal Learning:

So far, the system has relied only on regular RGB images (standard colour photos) to detect screws. However, in complex real-world conditions, just using visual data may not always be enough. For example, screws might be hidden under dirt, surface patterns might confuse the model, or poor lighting could reduce visibility.

Future work could explore multi-modal learning, which simply means combining information from multiple types of sensors. For example:

- Infrared cameras could detect hidden screws by picking up heat patterns or surface differences invisible to normal cameras.
- Depth sensors which essentially measure how far away objects are could help the system tell if a small bump is actually a screw head or just part of the surface.

Combining different types of sensor data„ a technique called sensor fusion, would likely make detection much more reliable, even under challenging conditions such as dark environments, glossy surfaces, or cluttered backgrounds. This would make the robotic disassembly system stronger, more flexible, and better able to handle a wider variety of real-world e-waste.

References

[1] United Nations Institute for Training and Research, "Global e-waste monitor 2024: Electronic waste rising five times faster than documented e-waste recycling," 2024, accessed April 29, 2025. [Online]. Available: https://unitar.org/about/news-stories/press/global-e-waste-monitor-2024-electronic-waste-rising-five-times-faster-documented-e-waste-recycling

[2] C. S. S. S. D. J. G. R. Rujanavech C, Lessard J, "Liam – an innovation story," September 2016, accessed April 29, 2025. [Online]. Available: https://www.apple.com/environment/pdf/Liam_white_paper_Sept2016.pdf

[3] G. R. F. A. Redmon J, Divvala S, "You only look once: Unified, real-time object detection," *arXiv preprint arXiv:1506.02640*, 2016, accessed April 29, 2025. [Online]. Available: https://arxiv.org/pdf/1506.02640

[4] UNSW Sites, "Australian laws lag on electronic waste management," 2016, available from: https://www.unsw.edu.au/newsroom/news/2016/07/australian-laws-lag-on-electronic-waste-management.

[5] M. Islam and N. Huda, "E-waste management practices in australia," in *Elsevier eBooks*, 2020, pp. 553–576, available from: https://doi.org/10.1016/b978-0-12-817030-4.00015-2.

[6] S. Atiya, Y. Chung, and G. George, "A comparative study on e-waste management systems in developed and developing countries," in *Practice, Progress, and Proficiency in Sustainability*, 2023, pp. 90–121, available from: https://doi.org/10.4018/978-1-6684-7573-7.ch006.

[7] E. Yildiz, T. Brinker, E. Renaudo, J. Hollenstein, S. Haller-Seeber, and J. Piater, "A visual intelligence scheme for hard drive disassembly in automated recycling routines," 2020, available from: https://doi.org/10.5220/0010016000170027.

[8] B. Mc Manus, “Why millions of usable hard drives are being destroyed,” 2023, available from: https://www.bbc.com/news/business-65669537.

[9] G. Foo, S. Kara, and M. Pagnucco, “Challenges of robotic disassembly in practice,” *Procedia CIRP*, vol. 105, pp. 513–518, 2022, available from: https://www.sciencedirect.com/science/article/pii/S2212827122000853?via%3Dihub.

[10] T. ID and A., “Ai vs. machine learning vs. deep learning vs. neural networks,” 2025, available from: https://www.ibm.com/think/topics/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks.

[11] E. Gadelmawla, “Computer vision algorithms for measurement and inspection of external screw threads,” *Measurement*, vol. 100, pp. 36–49, 2016, available from: https://doi.org/10.1016/j.measurement.2016.12.034.

[12] Z. Wang, H. Li, and X. Zhang, “Construction waste recycling robot for nails and screws: Computer vision technology and neural network approach,” *Automation in Construction*, vol. 97, pp. 220–228, 2018, available from: https://doi.org/10.1016/j.autcon.2018.11.009.

[13] S. Mangold, C. Steiner, M. Friedmann, and J. Fleischer, “Vision-based screw head detection for automated disassembly for remanufacturing,” *Procedia CIRP*, vol. 105, pp. 1–6, 2022, available from: https://doi.org/10.1016/j.procir.2022.02.001.

[14] C. Rujanavech, J. Lessard, S. Chandler, S. Shannon, J. Dahmus, and R. Guzzo, “Liam - an innovation story,” 2016, available from: https://www.apple.com/environment/pdf/Liam_white_paper_Sept2016.pdf.

[15] M. Heffernan, “Apple’s iphone recycling robot shifts gears,” 2024, available from: https://resource-recycling.com/e-scrap/2024/06/13/apples-iphone-recycling-robot-shifts-gears/#:~:text=First%2C%20a%20worker%20feeds%20a,cooling%20chamber%20to%20freeze%20it.

[16] S. Kolan, “What is tesla vision? how tesla trains deep neural nets and how you can learn that! (top courses),” 2021, available from: https://www.youtube.com/watch?v=dkNvMkSmU9E.

[17] Recycling Near You, “Planet ark’s recycling near you. computers,” available from: https://recyclingnearyou.com.au/computers/#:~:text=Computers%20can%20be%20dropped%20off,%2C%20hard%20drives%2C%20and%20motherboards.

[18] Cleanaway, “Secure e-waste destruction,” 2022, available from: https://www.cleanaway.com.au/waste/secure-e-waste-destruction/.

[19] Ecocycle, “How do i safely recycle old hard drives?” 2023, available from: https://ecocycle.com.au/how-do-i-safely-recycle-old-hard-drives/#:~:text=The%20safe%20way%20to%20recycle,for%20old%20computers%20and%20TVs.

[20] M. Dupont, “History of yolo: From yolov1 to yolov10,” 2024, available from: https://www.labelvisor.com/history-of-yolo-from-yolov1-to-yolov10/.

[21] C. N, “Screw detection dataset,” 2023, accessed April 30, 2025. [Online]. Available: https://universe.roboflow.com/university-of-limerick/screw_detection-ccbng

[22] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” 2015, available from: https://arxiv.org/abs/1505.04597.

[23] J. Torres, “Yolov8 architecture; deep dive into its architecture -yolov8,” 2025, available from: https://yolov8.org/yolov8-architecture/.

[24] C. N, “How apple’s daisy de-manufacturing machines battle e-waste,” 2023, accessed April 30, 2025. [Online]. Available: https://www.wallpaper.com/tech/how-apple-daisy-de-manufacturing-machines-battle-e-waste