

# Digital Twin Framework and Auto- Linking for Management of Legacy Assets

by

Chloe Edwards

A thesis  
presented to the University of Waterloo  
in fulfillment of the  
thesis requirement for the degree of  
Masters of Applied Science  
in  
Civil Engineering

Waterloo, Ontario, Canada, 2021

© Chloe Edwards 2021

## **AUTHOR'S DECLARATION**

I hereby declare that the contents of this thesis were of both my sole authorship and co-authorship. The Statement of Contributions dictates the chapters which include co-author contributions. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners. I understand that my thesis may be made electronically available to the public.

## **STATEMENT OF CONTRIBUTIONS**

Chapters 5 and 6 are the product of a collaborative research effort with Daniel Lopez. The research was conducted at the University of Waterloo as part of the OPG Digital Twins for Infrastructure project.

## **Abstract**

The integration of Digital Twins (DTs) for facility and asset management is becoming increasingly common with the advancement of sensor and modelling technology. A digital twin is defined as a responsive and virtual representation of a physical system or sub-system. While traditionally used at the design stage, there are many benefits to incorporating this technology during operation and maintenance of assets. The application area examined in this thesis relates to Nuclear Power Plant (NPP) facilities with focus on legacy assets. Integration of DTs for facility management (FM) of NPP facilities should improve the efficiency of many processes involved in operations and maintenance of assets. DTs are also advantageous in this application due to their use of virtual representation that can be used for training, scenario modelling, and problem solving with minimal asset down time. To streamline the integration of DTs into current FM standard procedures, the development of a comprehensive DT framework that defines levels of DT development within each NPP subsystem, identifies workflows, and details support documents is required. This thesis aims to develop a DT framework that can be applied to varying levels of legacy assets within an NPP. The framework will guide operators and engineers through the decision-making process that is involved when implementing DTs.

The presented DT framework incorporates a layered approach to demonstrate the connections between the physical environment and the virtual digital twin. The underlying levels include perception, communication, and action, each of which represent individual processes that contribute to a functional virtual model. This DT framework uses imagery, 3D scanning, operational data, and asset management software in an organized manner to streamline information integration for digital twin construction. Most of the technology and software included in the framework are commonly used in current practices, which reduces the initial cost and training investment. Since the DT framework presented was specifically developed for NPP FM application, the levels of DT were redefined to accommodate existing NPP FM practices. The support documents and workflows were also adapted according to current practices to simplify the employment of this technology.

Implementing DTs within the presented application of a legacy NPP requires a high level of effort that includes digitizing records, creating models of equipment and processes, and linking these digital representations in a functional way within the overall framework. Some of this effort can be automated to improve overall efficiency and reduce time efforts. Auto-linking was developed as a

tool that automates steps in the process of identifying physical asset labels within an image and 3D scan of a space within the facility including multiple physical assets such as mechanical, electrical, plumbing equipment, valves, instruments and other related assets. Those asset labels, the information they contain, and the 3D point cloud segments and images related to each tag's asset must be linked to their respective asset's information in an asset database that is used for NPP asset management. The 3D point cloud segments, images, and digital database serve as part of the foundation for the NPP DT being constructed within the framework presented. The Auto-linking algorithm uses a deep-learning based approach to object (tag) detection, employs optical character recognition (OCR) to read the asset tag's unique identifier, then sorts this information into a format that directly relates it, via the asset database, to each 3D point cloud of each scene in which the asset exists. Through Auto-linking, the procedure of labelling legacy assets throughout the facility and documenting existing assets for a digital database is accelerated compared to the current manual procedures.

This thesis identifies the challenges faced in FM for NPP facilities and legacy assets and demonstrates how and where DTs could be implemented for improved efficiency. The primary contributions of this thesis include the framework and support tools that can be used for some tasks required for implementing DTs for NPP FM, as well as the Auto-linking algorithm for automated asset management database development or enrichment. While these tools were developed specifically for NPP application, other application areas and adaptation for future work is also highlighted throughout the thesis.

## Acknowledgements

I would like to begin by expressing my sincere gratitude for the support and encouragement of Dr. Carl Haas throughout the entirety of my graduate studies and research. The continuous academic and personal guidance from Dr. Haas has helped me grow as both a researcher and an individual. I would also like to acknowledge Dr. Sriram Narasimhan and Dr. Giovanni Cascante for their support and guidance throughout the course of this research project. I have greatly appreciated the work environment created by these advisors that encouraged new ideas, open discussions of interesting concepts, and confidence in sharing opinions.

This research was made possible by the collaboration with Cristobal Lara and Daniel Lopez. I would like to thank them for this collaboration and the advice they provided that improved this research. Throughout the course of this research, I have learned a lot from Cristobal and Daniel with respect to their research areas and appreciate their contribution to the success of this project.

I would like to acknowledge Ontario Power Generation (OPG) for funding this research. Our industry partnership has provided notable insight into the current procedures and practices of Nuclear Power Plant facility management that greatly contributed to the research presented. I would specifically like to thank Wilson Chow, Derek Ellie, Jaydev Chauhan, Mike Di Lisi, Richard Sunnucks, and Derek Bilyk, at OPG for the many hours spent in team meetings, demonstrating procedures, gathering data, and hosting site visits.

Lastly, I extend a heartfelt thank you to my family and friends who have cheered me on throughout this academic endeavor with unwavering love and support.

## Table of Contents

AUTHOR'S DECLARATION .....	ii
STATEMENT OF CONTRIBUTIONS .....	iii
Abstract .....	iv
Acknowledgements .....	vi
List of Figures .....	x
List of Tables .....	xii
List of Acronyms .....	xiii
Chapter 1 Introduction.....	1
1.1 Problem Statement .....	2
1.2 Background .....	2
1.3 Research Objectives and Contribution .....	3
1.4 Scope .....	3
1.5 Methodology .....	3
1.6 Thesis Organization.....	4
Chapter 2 Literature Review .....	5
2.1 Facility Management trends in Nuclear Industry .....	5
2.1.1 Industry 4.0.....	6
2.1.2 Digital Twin Technology .....	7
2.1.3 Building Information Modelling .....	8
2.2 3D Technology for Asset Management.....	9
2.2.1 Scanning Technology and Point Clouds.....	10
2.3 Object Detection.....	10
2.4 Research Gaps based on Literature Review .....	12
Chapter 3 Proposed Methodology .....	15
3.1 Experimental Framework Development.....	15
3.1.1 Literature Review Contribution.....	15
3.1.2 Industry Participation .....	16
3.1.3 Experimental Analysis.....	17
3.2 Determination of Factors Affecting DT for Legacy Assets.....	17
3.3 Determination of Software and Technology Trends .....	18
Chapter 4 Proposed Framework for Implementing DTs for Legacy Assets .....	19

4.1 Description of Legacy Assets and Application .....	19
4.1.1 Structures, Systems and Components.....	19
4.2 Scope of Framework.....	20
4.3 Levels of Layered Framework.....	20
4.4 Defining Levels of a DT.....	23
4.4.1 Supporting Documents for Defining DT Level.....	25
4.5 Proposed Task Breakdown Structure for DTs.....	26
4.6 Using the Framework and Support Tools.....	28
Chapter 5 Auto-linking for a Digital Database of Legacy Assets .....	30
5.1 Definition of Auto-linking and Description of Applications.....	30
5.2 Requirements and Methodology.....	32
5.2.1 Object Detection.....	34
5.2.2 Machine Learning-Based Approaches.....	34
5.2.3 Deep Learning-Based Approaches .....	42
5.2.4 Implementing OCR to Extract Equipment ID .....	46
5.2.5 Mapping from Pixel Coordinates to 3D Point Cloud Coordinates .....	48
5.2.6 Linking Detections to Asset Management Database.....	49
5.2.7 Validation of Database .....	49
Chapter 6 Experimental Analysis of Auto-linking .....	50
6.1 Experimental Design .....	50
6.1.1 Current Procedure for Tagging Assets .....	50
6.1.2 Scope of Auto-linking Experimental Analysis.....	53
6.1.3 Assumptions and Limitations .....	54
6.1.4 Description of Process, Tools, and Software.....	55
6.1.5 Evaluation Metrics.....	59
6.1.6 Results of Auto-linking Analysis .....	60
6.1.7 Correction of Auto-linking OCR.....	69
6.1.8 Mapping Detections to Point Coordinates.....	70
Chapter 7 Discussion.....	72
7.1 Evaluation of the DT Framework for NPPs .....	72
7.1.1 Limitations of DT Framework.....	73
7.2 Evaluation of Auto-Linking .....	74



7.2.1 Combination of Auto-Linking and Human Operator .....	75
7.2.2 Limitations of Auto-Linking Algorithm.....	75
Chapter 8 Conclusion and Future Work.....	77
8.1 Future Work .....	78
References .....	80
Appendix A Machine Learning-Based Approaches.....	85
Appendix B Deep Learning-Based Approach.....	111
Appendix C Database of Existing Equipment ID Tags in Test Dataset.....	125
Appendix D Output from Auto-linking Algorithm containing GeoTag Information.....	139
Appendix E Python Script for Correcting OCR Detected Text.....	144

## List of Figures

Figure 4.1 A layered framework to demonstrate the tools and concept that make up a virtual representation of a physical asset .....	21
Figure 4.2 The five levels of DT as they relate to application in the NPP industry for legacy assets and SSCs .....	24
Figure 4.3 This task breakdown structure is a support tool that identifies the tasks that should be completed for decision making related to implementation of DTs for a specific SSCs.....	27
Figure 4.4 Example of a branch of a systems map to understand the related systems and sub-systems of an SSC.....	29
Figure 5.1 Auto-linking pipeline .....	30
Figure 5.2 Workflow for Auto-linking in OPG Asset Management Procedure .....	32
Figure 5.3 Auto-linking pipeline developed for automation of digitizing the SSC database and asset management procedure.....	33
Figure 5.4 Pipeline for SIFT approach demonstrating the matching of features between a target image and a scene image.....	35
Figure 5.5 Example of a tag that was detected using SIFT approach.....	36
Figure 5.6 The pipeline for the MSER approach to show how the equipment ID tag is detected using MSER features .....	36
Figure 5.7 MSER regions detected in an overall scene to be used for tag detection.....	37
Figure 5.8 Pipeline to demonstrate the process of using the Hough Transform for equipment ID tag detection .....	37
Figure 5.9 The pipeline related to the White Regions approach which processes images and sorts them based on a set of baseline parameters to detection potential equipment ID tags.....	38
Figure 5.10 Example of the white regions approach which identifies white regions within the scene images then bounds potential equipment ID tags.....	39
Figure 5.11 Examples of each approach using the same scene image .....	40
Figure 5.12 The YOLOv3 network architecture (Kathuria, 2018).....	43
Figure 5.13 The feature extractor known as Darknet-53 that is used in YOLOv3 (Redmon & Farhadi, 2018).....	44
Figure 5.14 Performance of YOLOv3 against other algorithms based on a common metric: COCO mAP-30 and inference time (Redmon & Farhadi, 2018) .....	45
Figure 5.15 Metrics that measure the quality of the YOLOv3 algorithm (Redmon & Farhadi, 2018)45	

Figure 5.16 Breakdown structure of OCR.....	47
Figure 6.1 Workflow of conducting the Auto-linking analysis to evaluation object detection and OCR output.....	54
Figure 6.2 Examples of the three classes. Left to right: Equipment Tag A, Equipment Tag B, and Equipment Tag C.....	56
Figure 6.3 The obj.names file that was used to identify object labels.....	58
Figure 6.4 The obj.data file for training and testing.....	58
Figure 6.5 Command prompt that runs the YOLOv3 training for custom dataset using pre-trained weights.....	59
Figure 6.6 Examples of detections for each class. (a) OPG Equipment Tag A detection, (b) OPG Equipment Tag B detections, (c) OPG Equipment Tag C detection and (d) Combination of OPG Equipment Tag A and OPG Equipment Tag B detections.....	62
Figure 6.7 Text files that correspond with the detections shown in Figure 6.6.....	63
Figure 6.8 Poor resolution as demonstrated by these images could impact the detector accuracy (a) IMG_1211.JPG and (b) IMG_1329.JPG.....	65
Figure 6.9 Instances where coverage causes an Equipment ID tag to be undetected. (a) IMG_2340.JPG contains one tag covered by equipment and another that was cut off by image boundary (b) IMG_1206.JPG shows a blank tag that was not detected.....	66
Figure 6.10 Undetected Equipment ID tags due to lighting (a) IMG_1288.JPG and (b) IMG_1247.JPG.....	67
Figure 6.11 Instances where Equipment ID tags are not detected due to damage to the object itself (a) IMG_1167.JPG shows damage to the text and (b) IMG_1242.JPG shows discoloration.....	67
Figure 6.12 Examples of when normal Equipment ID tags are undetected despite lack of visual discrepancy to the standard format (a) IMG_1205.JPG and (b) IMG_1241.JPG each have one false negative.....	68
Figure 6.13 Demonstration of Auto-linking from Two Scan Setups.....	71

## List of Tables

Table 5.1 The advantages and disadvantages of each machine learning-based approach based on the experimental test setup .....	40
Table 6.1 Description of the tab options that are used in the asset management tool TruView.....	51
Table 6.2 Confusion matrix for the Auto-linking object detection .....	64
Table 6.3 Possible reasons for undetected Equipment ID Tags and Comparison to Total Values .....	68
Table 6.4 Comparison of Original OCR Text to Corrected and Actual Text.....	69
Table 6.5 Relevant Information for Detected Equipment ID Tags .....	71

## List of Acronyms

AEC	Architectural, Engineering, and Construction
AI	Artificial Intelligence
AR	Augmented Reality
BIM	Building Information Model or Building Information Modelling
CC	Criticality Code
CNN	Convolutional Neural Network
COG	CANDU Owners Group
CPS	Cyber Physical Systems
DT	Digital Twin
FM	Facility Management
GPU	Graphics Processing Unit
IAEA	International Atomic Agency
IoT	Internet of Things
LOD	Level of Detail
NPP	Nuclear Power Plants
O&M	Operations & Maintenance
OCR	Optical Character Recognition
OPG	Ontario Power Generation
PCA	Principal Component Analysis
SIFT	Scale Invariant Feature Transform
SSCs	Systems Structures and Components
VC	Virtual Commissioning
VR	Virtual Reality
YOLO	You Only Look Once

# Chapter 1

## Introduction

The popularity of automation and digital technologies in the Architectural, Engineering, and Construction (AEC) industry is growing rapidly with the advancement of available software and technology. This has specifically become popular with Digital Twins (DT), Building Information Modelling (BIM) and other Cyber-Physical Systems (CPS) for management of design, construction, and operation of infrastructure projects. Along with the advancement of these technologies there is an increasing need for changes to current standards and procedures. The digitization of infrastructure projects using DT, BIM and CPS has been more commonly used for new design and construction projects since they provide accurate and updated representation of as-built conditions which can be compared to the archived or design documents. DT, BIM, and CPS also provide useful 3D modelling that allows engineers, designers, or operators to understand potential design or construction challenges ahead of implementation. As such, the development of frameworks and system architectures for implementing these technologies have primarily focused on new design and construction projects. The lack of system architecture that focuses on using DT, BIM, or CPS for facilities with legacy assets has delayed the implementation for applications such as asset and facility management for infrastructure currently in use.

Despite the lack of support frameworks, there are many advantages to implementing DT, BIM, or CPS in facilities with legacy assets. The advantages range from improvements in productivity and efficiency of creating online databases of existing assets to using 3D models for predictive and preventative maintenance which contribute to increased asset life. These advantages are particularly useful for facilities such as Nuclear Power Plants (NPPs) because of the high quantity of assets and the critical nature of the systems and sub-systems. This thesis delves into the implementation of DT and automation for legacy asset management with a focus on NPPs. The framework developed outlines the layers that contribute to an operational DT for various management purposes. The support tools developed provide guidance to operators and engineers who will work with the DTs to improve common practices and procedures. In addition to the DT framework, this thesis presents Auto-linking as a tool that automatically detects assets and links essential information to an asset management tool. This tool was developed as part of the objective to digitize and automate processes that are involved in facility management construction or development.

## **1.1 Problem Statement**

The popularization of incorporating digital models for assets, infrastructure, and facilities plays a large role in the advancement of Industry 4.0. The fourth industrial revolution, known as Industry 4.0, focuses on the inclusion of automation, computing, and technological advancement of existing manufacturing and industrial norms. The use of DT, BIM and other 3D technology is an outcome of Industry 4.0 that is commonly seen in the AEC industry. There are many applications for this technology including understanding the behaviour of complex systems and infrastructure under various environments, better predictions for useful asset and material life, improved custom solutions to complex issues, and advanced training or onboarding. The benefits of using semantically rich digital models are primarily focused on design, building monitoring, system analysis, and operations. There appears to be a shortage of example use cases where DTs or BIM are incorporated into the management of use-phase facilities and legacy assets. This limitation could be attributed to the lack of a framework or standard architecture to direct implementation of DTs for legacy assets.

The lack of a standard framework or system architecture related to DTs for NPP facility management applications has led to challenges for implementation. The main challenges faced include uncertainty in software and technology selection, defining appropriate levels of detail or development for DTs, and maintaining a standard procedure for implementing DTs. The development of a comprehensive framework that covers definitions of DTs, description of applications, and a breakdown of workflows would streamline the integration of DTs and digital technology for facility management. The framework would also ensure consistent practices throughout facilities, which allows for efficient management and operations. There is a clear need for a system architecture or framework that would eliminate these challenges such that NPP facilities can implement DTs and automatic processes to their full potential.

## **1.2 Background**

The positive impact and benefits to integrating digital technologies for asset management are evident in ongoing research, case studies, and from an industry perspective. The use of automation has also been shown to produce higher productivity and efficiency for common tasks involved in facility management. Along with increased productivity, digitization of assets can improve functionality and overall useful life of the asset by identifying technical issues and simulating potential solutions in a preventative manner. These benefits have been demonstrated primarily through design and new construction projects where early phases of projects are modelled digitally for comparison with the

as-built model. Despite the lower rate of implementation for legacy assets, there are comparable benefits when using DT, BIM, or CPS for facilities that are within the operations and maintenance phase.

### **1.3 Research Objectives and Contribution**

The primary research objective presented in this thesis is the development of a comprehensive framework for implementing DTs for management of legacy assets in NPP facilities. The framework includes multiple support tools for smooth integration into the current practices exhibited in the industry. The secondary objective presented in this thesis is the development of Auto-linking, a tool that was devised to automate the process of developing a digital database of assets. This tool is introduced as a method for computerization of assets which is a vital layer of the overall framework. The main contribution of the research presented is a demonstration of the framework and Auto-linking as an asset management tool for NPP facilities.

### **1.4 Scope**

The concepts and ideas presented in this thesis were developed with the intent of application for NPP facilities. The research was conducted as an industry-funded project that aimed to develop a framework and system architecture for the use of DT in NPP facilities. The scope of this research is as follows:

- Conduct a review of current practices and approaches for asset management in NPP facilities. This included a thorough literature review and collaboration with industry.
- Develop strategies for integrating DT and automation into the current practices for improved efficiency as well as time and cost savings.
- Provide support tools, definitions, and structures to identify the process of implementing DTs for legacy assets.
- Demonstrate the use of Auto-linking for asset management through an analysis based on industry provided data.

### **1.5 Methodology**

The research approach that was used to develop the framework, support tools, and Auto-linking algorithm primarily involved literature review, industry participation, and experiments. The literature



review indicated common practices and trends as well as ongoing research related to the field. Industry partnership provided insight into current procedures, complexity of the assets, systems and sub-systems and provided constructive feedback while developing the framework. Once the concepts were consolidated, an analysis was conducted to develop, experiment with, demonstrate and evaluate the Auto-linking algorithm. Chapter 3 includes an in-depth outline of the methodology.

## **1.6 Thesis Organization**

The organization of this thesis is described below:

Chapter 2 provides an in-depth literature review that covers topics including facility management trends, DT technology, BIM, Industry 4.0, 3D technology and software. The chapter concludes with a summary of the identified research gaps.

Chapter 3 covers the proposed methodology in detail, explaining the contribution of the literature review findings as well as the industry participation. It later discusses how important factors and software requirements were determined.

Chapter 4 details the proposed framework for developing DTs for legacy assets. This chapter contains the support tools and multiple figures to illustrate the process of DT implementation.

Chapter 5 focuses on the Auto-linking concept that was developed as a method to construct and connect digital assets to asset management tools.

Chapter 6 covers the experimental analysis of Auto-linking which was completed using data provided by industry partners. The steps involved in carrying out the Auto-linking process are discussed in detail. The results are evaluated based on a specified set of metrics.

Chapter 7 is a detailed discussion of the framework development and Auto-linking analysis evaluations.

Chapter 8 concludes this thesis with an overview of the research conducted, analysis, and a summary of the limitations involved. This chapter concludes with an explanation of the research contributions as well as recommended future work related to this thesis.

## Chapter 2

### Literature Review

#### 2.1 Facility Management trends in Nuclear Industry

The continuous improvements in automation, sensing technologies, and computer vision has led to imperative trends in facility management that aim to increase the productivity and efficiency for facilities within the Nuclear industry. These trends mirror the definition of Industry 4.0 due to the incorporation of Cyber-Physical Systems (CPS), including Digital Twin (DT) technology, Building Information Modelling (BIM) and use of other 3D modelling technology. Traditionally, the use of these digital modelling methods has been incorporated during early project stages to monitor assets through design, construction, use, and end-of-life phases. However, literature suggests that incorporation of digital modelling for legacy assets would exhibit many of the same benefits if implemented during later stages of their use. Although the design stage is not included in modelling for legacy assets, the use of technology for complex systems can still provide real-time monitoring and semantic information for predictive and preventative maintenance (Yuhan, Chimay, & Weisheng, 2019). Given the drive and potential for reliable modelling to monitor complex systems and detect potential issues or failure, there is a need for dependable framework to guide employment at later stages. Recent works in the Architectural, Engineering, and Construction (AEC) Industry present conceptual frameworks and architectures for digitization of physical assets and related inventories.

Xu et al. define the integration of technology and computing in facility management as ‘cognitive facility management’ and provide a corresponding framework to describe the relationship between the physical and virtual environments (Xu, Lu, Xue, & Chen, 2019). The presented framework is comprised of eight layers including: environment, perception, data, communication, computation, application, action, and evaluation. Each layer describes vital contributions to retrieving information from the real-world system, understanding the information, communicating, and storing data in a virtual system, and lastly, applying the information in a facility management environment. The delimitation of layers for this virtual facility management system are necessary for users to understand the key processes in creating a digital model for asset management. Similarly, other authors have discussed the need for a framework that connects perception, network communication, and application for the purpose of a complete management-support system (Yuhan et al., 2019).

This section is intended to delve into the current practices of Industry 4.0, DT technology, BIM, and 3D technology for asset management as they relate to application in the nuclear industry. Each subsection will detail specific trends as they are related to legacy assets to identify existing architectures, frameworks, and current gaps in practices.

### **2.1.1 Industry 4.0**

Industry 4.0 is known as the fourth industrial revolution which involves a transformation into the digital world. A digital transformation for companies allows for direct collaboration and more accessible management through the entire lifecycle of a project. The primary drawbacks and reluctance to Industry 4.0 is the anticipated financial and time investment that is required for digitization of assets and processes (Ardito, Petruzzelli, Panniello, & Garavelli, 2018).

Examples of Industry 4.0 in the AEC industry includes but is not limited to the use of robotics, automated systems, Augmented Reality (AR), Artificial Intelligence (AI), DT, BIM and digital sensing technologies. A main advantage of incorporating robotics and automated systems is the positive impact it has on overall productivity of labor-intensive tasks, which is common in the AEC industry (Delgado et al., 2019). In addition to the increase in productivity, the use of robotics and automated systems have been proven to minimize work related injuries and stresses. It is evident that there are many advantages to adopting robotics and automated system, however Delgado et al. suggest that the reason for lack of adoption in the construction industry is related to a lack of research that identifies the challenges and applications. As such, the following categories of robotic, automated, and autonomous systems in the construction industry were investigated by Delgado et al.: off-site automated prefabrication systems, on-site automated and robotic systems, drones and autonomous vehicles, and exoskeletons. A study that involved perspectives from contractor-side, client-side, technical-side, and business-side of the industry was conducted to determine common limiting factors to integration of Industry 4.0 practices. Some of the main limiting factors identified included high initial capital investment, absence of the need to improve productivity, untrained workforce, lack of government incentives, easy access to labour, and a general aversion to change (Delgado et al., 2019). This study demonstrated how the low adoption of Industry 4.0 within the AEC field is not caused by a lack of applicable technologies or uses but rather due to a lack of belief that these technologies will have a significant and necessary advantage compared to traditional practices. Additional studies had been conducted that report similar findings of the lack of willingness to incorporate new technology in current practices.

Although there is evidence of slow adoption, a particular area of the AEC industry that is advancing with technology is the inclusion of augmented reality (AR) and virtual reality (VR) for management of building and facility assets. Studies have shown that the use of AR or VR could improve management efficiency by detecting assets for inventory, information retrieval, and on-site updating (Corneli, Naticchia, Carbonari, & Bosché, 2019). The rise in this technology is increasing with the use of BIM for infrastructure project but is also exhibiting delayed adoption due to risk of the high initial time and cost investment. It is evident from the studies available that the primary challenge of introducing technology to improve AEC industry practices is proving the advantages and benefits outweigh the initial costs and additional training. Current research could be expanded to demonstrate not only the applicability but also the transferability of practices to include robotics and automated systems.

### **2.1.2 Digital Twin Technology**

As mentioned in the previous sections, DTs have become increasingly popular within the AEC industry to model projects from the design phase through end-of-life phase. There is a high range of applicability of DTs with implementation in infrastructure, energy, transportation, and water markets. Although companies have been using digital models and simulations of systems for decades, the term ‘digital twin’ originated in 2003 by Michael Grieves and can be defined as “a responsive system connected between the physical and digital systems” (Arup, 2019). The definition has since evolved with the advancing technology and application of DTs. Digital twins were originally known to simply describe geometric properties of a physical asset in a three-dimensional virtual space (Grieves & Vickers, 2017). Development of simulation technology has therefore allowed for advancement of DTs to include the modelling of system forces and processes to evaluate operating behaviour over an asset’s life span. The modelling and simulating of this behaviour can provide insight to emergent, undesirable, or unpredicted behaviour to prevent and mitigate serious problems. Modelling this behaviour prior to the popularity of DTs was primarily conducted through expensive prototypes. When comparing the projected costs of the two methods, it is evident that the costs for a physical prototype will continue to grow in the future whereas the costs for virtual modelling exhibit a decreasing exponential projection. (Grieves & Vickers, 2017)

The financial benefits related to DTs has been demonstrated in Virtual Commissioning (VC) projects, which employ virtual modelling such as DTs to validate and verify systems. VC is done through either simulation or emulation of a complex system. Examples of using DT in VC shows that

the initial cost and time investment is paid off due to a reduction in real commissioning and maintenance costs. In addition to the financial advantage, the use of these techniques assists in training operators, support design of a system, and optimize performance and behaviour. (Ayani, Ganebäck, & Ng, 2018)

As the use of DTs become more prominent in refurbishment, commissioning, and management of complex systems, the implementation for legacy assets and decommissioning continues to grow. For example, CANDU has indicated a need for research in plant information modelling (PIM) that use digital twins for entire plant lifecycle visualization and monitoring (CANDU Owners Group, 2019).

In parallel to the current research in advantages of DTs for various applications, there have been developments in the approach for data linking related to DTs. The use of an integrated platform to continuously update and exchange operating data will improve the model accuracy throughout the design and use phases. Deep-Lynx is a data warehouse that stores user defined data that can retrieve requested system or sub-system information based on operator specified queries (Darrington, Browning, & Ritter, 2020). A digital warehouse that incorporates operator defined data improves the current manual nature of data exchange by providing digital access to information for operations and modelling. This type of data exchange tool can be applied to DTs for linking data and improving the accuracy of the input data for better modelling productivity and accuracy. The Deep-Lynx tool is an example of how digital access to data can create an integrated network for improved operability of DTs and other cyber-physical systems used for design and operations.

### **2.1.3 Building Information Modelling**

Building Information Models (BIM) are a form of digital representation that is commonly used for management, design coordination, drawing generation, and quantity take-off for civil infrastructure and buildings. Since its introduction in the 1990's BIM has become a reliable method of information exchange throughout the entirety of a project, because it prevents loss of information and increases productivity (Borrmann, König, Koch, & Beetz, 2018). BIM provides useful visualization of infrastructure through design and construction phases and can also provide enriched information to assist in operating and maintenance (O&M) phases (Lu, Xie, Parlikad, Schooling, & Konstantinou, 2020). Not only does it contain geometric properties of space, but the objects that are modelled are often associated with semantic information including component type, materials, technical properties, costs, and relationships with other objects (Borrmann et al., 2018). The amount of information

available in a given BIM is often defined by its ‘Level of Development’ (LOD), which describes the extent of detail and reliability.

While BIM has been an effective tool for project planning, design, and construction there are several limitations when adopting BIM for facility and asset management. Despite the efforts to further develop BIM-based asset management the following limitations and gaps are noticed in the O&M phases: lack of technology and integration with systems, lack of a framework for organizing information and specifying information, inadequate organization of BIM processes and workflows, and lastly the gap in recognized standards to maintain consistency in the field. (Lu et al., 2020)

The main course of addressing these limitations is the development of technologies that can be used in parallel with BIM. In particular, recent advancements in the Internet of Things (IoT), cloud computing, and sensing technologies allow for more attainable digitization and modelling of large facilities (Su, 2017). An example of advanced technology to improve BIM application is the use of inverse photogrammetry and BIM modelling to automatically label construction images (Braun & Borrmann, 2019). The algorithm presented by Braun et al. is useful for tracking progress of construction sites, yet the concepts presented are also useful for tracking and inventory of assets in facility management. The application of BIM in facility management could be further developed with the addition of a comprehensive and strategic framework to guide BIM-enabled facility and asset management. Along with a strategic framework, it is suggested that a clear definition and a practical process of gathering data for implementation should be derived and constant throughout industry (Pishdad-Bozorgi, Gao, Eastman, & Self, 2018). Pishdad-Bozorgi et al. presented a ten-layered framework that can be integrated into future research for the purpose of BIM-enabled facility management. This framework systematically identifies the need and objective of a project, data and compatibility, procedure for BIM modelling and quality control and assurance. This need for a systematic approach and clearly defined framework is echoed by the common challenges faced in the industry. Having a framework to guide operators in implementing this tool could address technological barriers and optimize the benefits of BIM.

## **2.2 3D Technology for Asset Management**

A key aspect of both DT technology and BIM is that they serve as a 3D digital visualization of the physical component or system. Gathering and registering 3D data has evolved in recent years with the advancement of sensing technology and devices. Common registration algorithms include Principal

Component Analysis method (PCA), Singular Value Decomposition (SVD) and Iterative Closest Point (ICP) (Bellekens, Spruyt, Berkvens, Penne, & Weyn, 2015). As the use of 3D models increased in popularity, automation of point cloud registration and semantic labelling of point clouds became prominent research topics to further improve the efficiency of point cloud creation.

### **2.2.1 Scanning Technology and Point Clouds**

A point cloud is an approach to visualizing and modelling the built environment through registration of digital 3D coordinates. Scanning technology is an approach to capturing the measurement data that is used to construct a point cloud. Methods of capturing 3D data include laser triangulation, structured light, photogrammetry, contact-based, and laser pulse-based 3D scanning technology. LiDAR is a method of 3D scanning through measuring the wavelength of last light being reflected off an object.

### **2.3 Object Detection**

Object detection is used in a wide range of applications to identify, classify and often locate certain objects within images and video. Computer vision techniques including machine learning and deep learning methods are often employed for object detection. The primary difference between machine learning and deep learning methods is that machine learning methods utilize defined features of the object being detected (Brilakis & Haas, 2019) whereas deep learning methods are typically based on convolutional neural networks that do not require specifically defined features. This section explores the use of both approaches for object detection with AEC applications. Common applications of object detection related to the AEC industry include the detection of barcodes and QR codes, license plate detection, railway signal detection, and construction management, among others. A common trait among these applications is that the objects are relatively standard. For example, barcodes and QR codes follow a rectangular or square shape and are black and white, license plates follow similar shape and colours, signals and symbols used in railway networks are standardized and construction assets are often similar across projects and sites.

The need to use computer vision techniques for object detection derives primarily from a need for more efficient and cost-effective approaches to detecting certain objects or assets for project management. Warsop et al. identified that a major challenge in asset management is the unreliability of documentation compared to the current state (Warsop & Singh, 2010). The infrastructure of focus in their work was railway networks, however this is a challenge that is faced in any large-scale infrastructure project or facility with updating or changing assets. The author declared that the use of

automation for detecting assets in the current state would increase the efficiency of the management procedure compared to manually detecting assets and changes. The method explored in this literature used a SIFT approach in both a template-based method and multi-resolution method. SIFT stands for Scale Invariant Feature Transform and searches an image for specific features to detect objects. The template based SIFT method uses phase correlation and matches the global appearance with image positions. The multi-resolution object recognition employed an edge-based recognition technique and trained detectors based on specified scales.

In a similar application, Stoitchkov et al. investigate the use of Convolutional Neural Networks (CNN) to detect railway equipment symbols in engineering plans to update dates asset management documents (Stoitchkov et al., 2019). This work was completed as part of the Railway Information Modelling research project (RIMcomb) to automate the recognition and identification of railway plan symbols then store the location and count data in a digital database. The process involved three primary steps, first to pre-process the images for training, then to train the CNN and lastly to detect the symbols by employing the CNN. To reduce computational effort, the images were cropped into Regions of Interest (ROI) such that the search area was reduced. The results of this analysis showed that the use of CNNs was a viable solution to reducing the time and effort of manually updating archived documents. The accuracy of these results was measured by determining a percent of the number of detections versus the actual number of symbols and secondly by comparing the location of the detection to the actual location of the symbol. (Stoitchkov et al., 2019)

Another common application of object detection for asset management is detecting barcode and QR codes within images and videos. Similar to the methods used for railway network symbols and objects, researchers have examined machine learning and deep learning approaches for barcode detection and classification. The machine learning and computer vision methods that were explored for barcode and QR code detection included corner detection, detecting regions of high x and y derivative differences, Maximal Stable Extremal regions for bar detection, and Hough Transform methods (Hansen, Nasrollahi, Rasmussen, & Moeslund, 2017). These techniques are useful for applications where the object of interest follows a standard shape and tone, such as barcodes and QR codes that are often black and white and either square or rectangular. However, the processing required to search images for these specified properties are computationally exhaustive unless the images is focused in on the barcode or QR code. Due to the high processing requirements of these methods, Hansen et al. suggest an approach using the YOLO algorithm with Darknet19. This



approach was trained on both 1D barcodes and QR codes with input size of 416x416 and proved to be a faster and more efficient detector compared to the machine learning approaches (Hansen et al., 2017). In addition to being computationally exhaustive, the use of traditional machine learning approaches for detection exhibited low performance due to variation in environmental conditions thus affecting image properties such as brightness or blur. The use of CNNs could address this drawback by training the network on thousands of objects under varying conditions. Chou et al. trained a six layer CNN architecture to detect partial QR-code patches based on frequency switches between black and white regions (Chou, Ho, & Kuo, 2015). Once the QR-code patch was detected, the Hough Transform was used to detect the boundaries of the QR-code. This is an example of reducing the computation power required by combining deep learning approaches with machine learning approaches.

Current trends in object detection show that both machine learning and deep learning approaches are viable options depending on the objective and application. Machine learning approaches that utilize algorithms such as SIFT, Hough Transform, Maximum Stable Extremal regions are favourable for detections that are within consistent environments with standard appearance. The primary advantages to the deep learning approaches that employ neural networks is the applicability for varying environments along with higher efficiency and speed of detection. The accuracy of detection appears to be largely based on the approach taken, the data used for detection, and the chosen neural network.

## **2.4 Research Gaps based on Literature Review**

This section explores the research gaps of key concepts related to this thesis based on the current state as described above. A general theme exhibited between Industry 4.0, DT technology and BIM is the slow development toward facility management including legacy assets in the AEC industry. Specifically, DT technology and BIM methods have been a staple for design and planning for decades. However, the advancement into later phases such as operating, maintenance, or decommissioning has not progressed as quickly. Many attribute this slow progression to high initial investment required to be spent on adapting new technology and the lack of a comprehensive framework for such application. Current trends related to Industry 4.0 and FM show that the adoption of digital and automated technology for asset management can improve the efficiency of large-scale facility management. Examples of improvements include digital replication of assets including visual representation and access to functional data that can be shared across teams within a company, virtual

training for onboarding processes, reduced exposure to potentially harmful environments through use of virtual twin monitoring, and virtual simulation of potential failure mechanisms, among others. To employ digital twin technology for asset management, it is necessary to develop a standard framework that details the contribution of technology, software, and planning considerations. A common trend in the literature review suggests a layered framework is a valuable tool for understanding individual contributions and the relationships between a physical asset and the corresponding virtual twin. A reoccurring theme is the inclusion of Perception, Communication, and Action layers which intend to represent the primary actions for constructing digital twins and virtual spaces for assets. The refinement of these layers based on standard practice followed in NPP facilities would streamline the implementation of digital twins for asset management.

Acquisition of 3D point data and 2D imaging of the physical environment is essential for constructing the digital database of assets in a facility. As discussed, there are various sensing technologies available to achieve a highly accurate digital representation of the assets in question. Laser scanners have become increasingly popular and easy to use for large objects such as buildings and large-scale assets. Advancements such as real-time registration and semantic labelling are continuously adding value to scan-to-BIM and 3D reconstruction of physical environments. The 3D point data and corresponding image data are used as a baseline for the digital twin modelling. Specifically, the point clouds provide geometric information for constructing a digital twin as well as location information which is useful for updating asset databases. The images provide updated representation of the asset in current state which also contributes to the asset database. Furthermore, image or video data can be used for detection of objects that would improve the efficiency of logging existing assets and determining their respective locations. Object detection has become increasingly prominent for asset management due to its benefits in updating documents in a fast-paced and accurate manner. Computer vision approaches such as machine learning, or deep learning are commonly used for this application of object detection and recognition. As such, research has begun to focus on incorporating object detection and localization to update archived documents and plan drawings, detect barcodes and QR codes for supply chain management, and determine vehicles or pedestrians in the automation industry. The applicability of object detection is varied yet the results are relatively consistent in showing an improvement in management productivity.

The takeaway from the current trends in research is that there remains a lack of implementation of these advancing technologies and management trends in large-scale facilities such as NPPs. Due to

the size of NPP facilities and complexity of their assets, there would be several benefits to incorporating digital twins and automated approaches to asset management. This thesis aims to address the gap of digital twin implementation for NPP application by developing a framework for standardized implementation and introducing an automated methodology for detecting assets.

## **Chapter 3**

### **Proposed Methodology**

This chapter describes the proposed methodology that was followed with the intent of facilitating the implementation of Digital Twins and automatic processes for the purpose of legacy asset management. This chapter includes an overview of the main contributors to the research as well as an explanation of the experimental analysis used to evaluate the concepts presented.

#### **3.1 Experimental Framework Development**

The proposed methodology described in this chapter was followed to develop a framework and procedure for improving the efficiency of facility management for legacy assets. Specifically, this research is focused on the process of implementing DT technology for systems and sub-systems in a Nuclear Power Plant (NPP). The literature review that was conducted guided the scope of the framework and identified specific application areas that require further research and development. In addition to the literature review, this research involved continual participation from industry partners that provided insight into common techniques, workflows, and challenges that are faced in monitoring and operating NPP systems and sub-systems. The automated processes developed, known as Auto-linking, which is presented as part of the framework was then evaluated based on an experimental analysis. This chapter also details the determination of factors that affect DT construction for legacy assets, and the selection of technology and software based on current practices in the industry.

##### **3.1.1 Literature Review Contribution**

A fundamental gap in the current NPP management trends is the lack of framework for implementation of DTs for legacy assets. Based on the literature review, a consistent pattern was that there are numerous sources of support for implementing DTs and similar technology such as CPS and BIM for management and monitoring of new construction or design of assets. The frameworks and support architectures presented for these applications frequently consisted of layered frameworks that identify relationships between physical and virtual environments. The application for legacy asset DTs would benefit from a layered framework to guide the perception of the physical environment, the communication between the digital representation and asset management tool, and lastly the performance of the DTs. The primary challenges that are faced when developing these frameworks for legacy assets as compared to new design include:

- Lack of updated design documents and plans that are needed for referencing when constructing the geometric model and physics model.
- High initial investment for the software and technology needed to obtain digital data of the physical environment.
- A misunderstanding of what defines a model as a DT and how to utilize the model to benefit and improve asset management procedures.

Given these challenges, it would be advantageous to redefine what DTs mean in terms of legacy assets to increase the applicability of DT technology. The main contributions of the literature review were, therefore, the exposure of existing gaps in current trends as well as inspiration for framework that will be proven useful in the AEC industry.

### **3.1.2 Industry Participation**

The participation of industry partners also played an essential role in understanding the challenges faced in managing and monitoring complex systems and sub-systems within an NPP. Collaboration with Ontario Power Generation (OPG) began in September 2019 with the objective of developing a system architecture for key NPP assets along with a decision support system based on current and emerging uses for DTs. This collaboration included bi-weekly meetings and interim progress presentations that were documented for reference and constructive feedback. The feedback provided assisted in maintaining practicality in the research to provide smooth integration with the current workflows and practices.

This industry partnership expressed an interest in improving the following aspects of their current NPP FM procedures and challenges:

- An approach to modelling systems and sub-systems such that operators and engineers can gain a better understanding of the systems and sub-systems operations to reduce downtime when an unexpected failure or issue arises. This approach should be applicable to various levels of systems and sub-systems throughout the NPP.
- An approach to improving the efficiency of their current digital database procedure. The existing procedure involves using 3D point clouds to provide virtual representation of the assets. Within these point clouds an operator or engineer manually labels assets according to their previously assigned unique identifiers. This point cloud also contain links that

relate to information such as ongoing maintenance work, related work documents, images, or other data.

The bi-weekly meetings were used to both inform industry partners of current research approaches as well as to strategize how the use of DTs could address the above listed concerns.

### **3.1.3 Experimental Analysis**

The overall framework and workflows that are presented in this thesis were not placed under experimental testing due to their conceptual nature. The experimental analysis presented in Chapter 6 corresponds to the auto-linking concept that is a major contribution of the DT framework. The setup, data acquisition, and methodology for the experimental analysis is explained in detail in Chapter 6. The data and software requirements were provided by OPG. The intent of conducting an analysis was to demonstrate the process of auto-linking, exemplify data input, limitations, and results. The analysis also demonstrates the advantages of automating the asset management processes and is a prime example of the role of computation in implementing DTs.

## **3.2 Determination of Factors Affecting DT for Legacy Assets**

As previously stated, the main contributors to gathering background information for DTs of legacy assets for NPP FM included literature review and collaboration with industry partners. Based on these sources, it was found that the following factors have an impact on the implementation of DTs for legacy assets:

- A comprehensive framework and work breakdown structure to guide operators when deciding to implement DTs.
- Defining the levels of DTs as they relate to assets and SSCs that are within the operating phase or nearing decommissioning phase.
- Time and cost investment for implementing DTs and using the technology to monitor issues or unpredicted failures of SSCs.
- Use of technology that can be adapted by operators without the requirement of prior expertise or extensive training.
- Visualization of the SSCs that can be used for training in addition to modelling.

### **3.3 Determination of Software and Technology Trends**

Different software and technology are used throughout the processes of digitizing the physical environment and assets into virtual replications. In addition to gathering the digital data, software is necessary to transfer data to usable formats and communicate with asset management tools.

According to the literature review and discussion with industry partners the following software and technology would be required for an asset for the implementation of DTs:

- 3D scanning technology to gather measurement and location data as 3D point clouds.
- HDR imagery for updated and high-resolution documentation of assets.
- Sensor technology for gathering and monitoring asset behaviour and operations.
- Registration software for registering and displaying point cloud data.

Specific software and technology recommendations are made in later chapters when relevant. The intent was to incorporate software and technology that is both accessible and simple to integrate into the existing asset management trends and practices. Incorporating software and technology that is already in use streamlines the training process and understanding of the functionality.

## Chapter 4

### Proposed Framework for Implementing DTs for Legacy Assets

This chapter introduces the proposed DT framework by first defining legacy assets and DT application, then outlining the scope of the framework, and lastly providing a detailed explanation of the framework and its components. When considering the implementation of DTs for NPP FM, it is necessary to first understand and define legacy assets in terms of NPP operation and maintenance. This chapter provides the appropriate background information prior to introducing the DT framework and support tools.

#### 4.1 Description of Legacy Assets and Application

A legacy asset is a term that can be applied to structures, systems, and components (SSCs) in an NPP that are well into their operational life. The implementation of DTs is highly advantageous for legacy assets despite the lack of current application and availability of implementation frameworks. DTs for legacy assets improve plant management by addressing safety, operation, and other service concerns when SSCs are experiencing unpredicted or undesirable issues. Through digital replication of an SSC, an operator can gain an understanding of the asset's behaviour under various stresses and external environments to detect the root of an issue. Modelling this behaviour allows for better planning and operation that will reduce delays or shutdowns in the plant. DTs also include virtual representation of the current state of the facility, which allows for an accurate database of legacy assets and corresponding work orders, maintenance, or operating concerns. This section defines SSCs in detail and discusses the applications for DTs in NPP facilities.

##### 4.1.1 Structures, Systems and Components

Assets within an NPP are defined as SSCs and are typically evaluated based on their level of criticality. Common categories of SSCs include (IAEA, 2015):

- 1) Critical irreplaceable SSCs: this type of SSC is typically passive, and failure could result in limiting the plant design life.
- 2) Critical but replaceable SSCs: this type of SSC is active and may be replaced during plant outages or through refurbishment procedures.
- 3) Less critical SSCs: this type of SSC is most common in a plant such as instruments and valves.



4) Non-safety items: this category is for isolated components that may become obsolete.

Categorizing SSCs according to level of criticality provides consistency when evaluating the state of SSCs and enforcing management plans. For instance, management of critical irreplaceable SSCs should aim to maintain the SSC such that it reaches its expected design life. This would require a thorough understanding of degradation and failure mechanisms, safety procedures, and technical specifications. While these SSCs require in-depth assessment, the lower critical categories could be managed and evaluated based on simplified analysis of behaviour and failure mechanisms. This methodology of categorizing SSCs in an NPP is also useful for planning the implementation of DTs since the DT of an SSC could be used for quality assessment and other management needs. Other parameters that contribute to categorizing SSCs and legacy assets include location, accessibility, and level of support documentation. The need for an understanding of an SSC's criticality, function, and location become apparent in later sections where the development of the framework is discussed.

## **4.2 Scope of Framework**

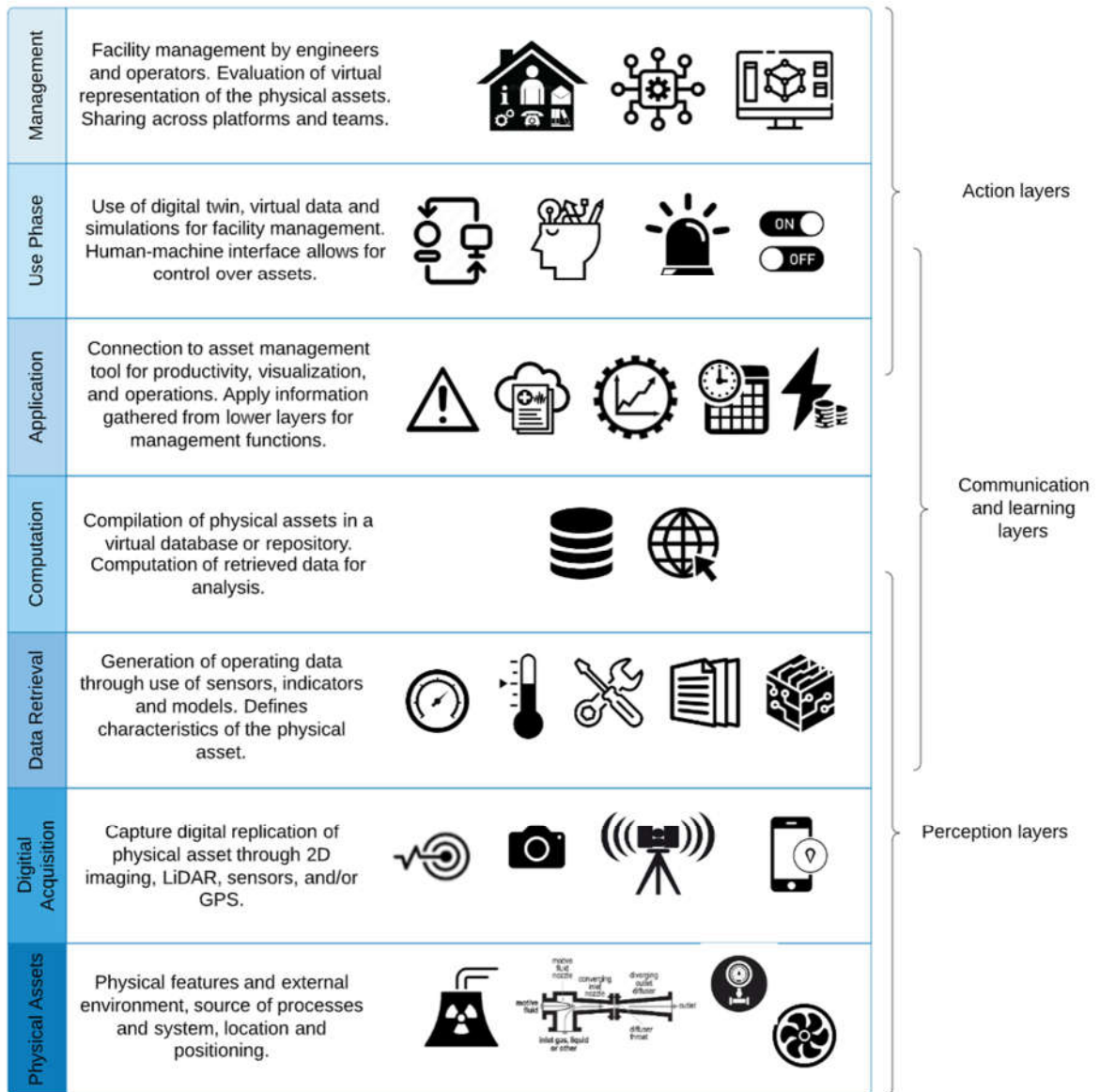
The primary intent of this framework is to provide support to operators and engineers who are implementing DTs for asset management of SSCs with specific focus on legacy assets. The scope of the proposed framework includes:

- A layered system architecture that defines the fundamental steps of modelling from physical environment to virtual asset tools.
- A support tool for defining levels of DTs and choosing the appropriate level dependent on the SSC parameters.
- A workflow diagram for implementing DTs for NPP FM.
- A methodology for organizing the networks between multiple DTs of SSCs throughout the NPP facilities.

## **4.3 Levels of Layered Framework**

The development of a comprehensive framework to support the implementation of DTs for NPP FM involved a multi-layered format to show the various tools and concepts that contribute to the final product. This layered framework is shown in Figure 4.1, which defines perception layers, communication and learning layers, and action layers as they apply to SSCs and legacy assets. The

structure of the layered framework shown was inspired by the research conducted by Xu et al who demonstrated how a layered approach can identify the critical contributions to digital facility management (Xu et al., 2019). Following the image are detailed descriptions of each layer.



**Figure 4.1 A layered framework to demonstrate the tools and concept that make up a virtual representation of a physical asset**

This general framework consists of seven layers, each contributing to the digitization of a physical asset. The significance of each layer is described below.

- The base layer is labelled “Physical Assets” to represent the SSC as it is in its existing physical environment. Notable features of the physical asset that should be documented and considered for implementation of DTs include location, accessibility concerns, sources of energy if applicable, dependent or related SSCs, as well as information related to failure modes and safety. Gathering this information will give an overview of the asset such that that operator can begin to understand what level of detail the DTs should represent. This relates to the critical categories in the previous section, which demonstrated how certain procedures should be followed based on the critical nature of the asset.
- The layer that follows is “Digital Acquisition” which represents the stage of capturing digital 3D replication of the physical asset through various methods. Digital replication might include all or some of the following: 2D imagery through digital cameras, point data such as LiDAR data using 3D sensing technology, and/or GPS. This layer is solely related to the visual aspect of the digital model, it provides a perception of the asset in its current state. The point clouds registered from the LiDAR data can also be used in later stages for geometric modelling and obtaining measurement data.
- The third layer is “Data Retrieval”, a perception and learning layer because it conveys important data about the physical asset. Operating data can be generated through the appropriate sensors installed in the physical environment (pressure, temperature, vibrational, and others). Data may also be generated from indicators or existing models of the physical asset. Retrieval of this data is essential for defining characteristics of the physical asset that will contribute to the asset health monitoring and management.
- Next is the “Computation” layer which is intended to compile the data generated from the previous data retrieval and data acquisition layers to compile a virtual database or repository of the physical asset. The software used for this layer may vary depending on application. The intent of the database is to use unique identifiers to sort and store all pertinent data related to a physical asset. The computation layer also functions as the layer that produces DT visual representation appropriate software. The provided support documents act as a tool that guides the decision-making process for determining the level of DT that should be implemented based on the SSC’s properties.
- The fourth layer is the “Application” layer to connect the information in the digital database and DTs to software tools that are used for asset management. This layer is multi-

functional as it is intended to connect the DTs and related digital information to tools that can be used for productivity, visualization, and planning. This layer is useful for accessing basic information about an asset, such as maintenance schedules, dimensions, location, and images. It also provides 3D visualization that can act as a virtual tour of the NPP.

- The next layer is, “Use Phase,” indicating the point of human-machine interaction and control over the digital assets. In this layer an operator can view and manipulate the DTs to simulate behaviour, model potential failure mechanisms, and adjust parameters to understand how the asset functions. This layer is most useful for operators and engineers that will use the DTs for preventative and predictive maintenance or repair. The DTs can also be employed as a training tool for new employees.
- The final layer is “Management” indicating that it is the combination of all lower layers to result in a cohesive system that can be used by engineers, operators, and managers for multiple purposes. The objective is to share the final digital product across platforms and teams within the organization.

#### **4.4 Defining Levels of a DT**

When implementing DTs for legacy assets it is necessary to pre-determine what level of DT would be applicable for the asset of interest. The levels of DTs dictate what functions the DT should inhibit and to what detail. In 2019, Arup published five key levels that aim to guide the development of DTs and define how they evolve from a simplified digital model to a DT with autonomous capabilities (Arup, 2019). To accommodate the advancing applications of DTs for legacy assets, these levels were refined according to common trends and practices of NPP FM, as shown below in Figure 4.2.

Level of DT	Definition <i>(These definitions are based on the definitions provided by ARUP and specified to the procedures at OPG)</i>
Level 1	A basic model of the real-world system with limited functionality and does not have learning or predictive ability. This level provides digital inventory without geometric modelling or feedback capabilities <ul style="list-style-type: none"> <li>• Point cloud</li> <li>• Auto-links to asset management</li> </ul>
Level 2	A model of the real-world system with limited feedback capabilities. Sensors that feed information to a human operator provide performance information that can be linked to asset management tool. <ul style="list-style-type: none"> <li>• Point cloud</li> <li>• Manual output sensors</li> <li>• Auto-linked to asset management</li> </ul>
Level 3	A digital model including the geometric and physics modelling that provides feedback for predictive maintenance. <ul style="list-style-type: none"> <li>• Point cloud</li> <li>• Geometric and physics model</li> <li>• Sensors</li> <li>• Auto-linked to asset management</li> </ul>
Level 4	A complete digital model that has the capability to learn from various sources of data and other components. Has the ability for autonomous decision-making <ul style="list-style-type: none"> <li>• Point cloud</li> <li>• Complete model</li> <li>• Sensors</li> <li>• Auto-linked to asset management</li> <li>• Predictive maintenance</li> </ul>
Level 5	A fully digital model capable of making autonomous reasoning on behalf of human operators <ul style="list-style-type: none"> <li>• Point cloud</li> <li>• Full model</li> <li>• Autonomous decision-making for performance</li> <li>• Auto-linked to asset management and other DTs</li> </ul>

**Figure 4.2 The five levels of DT as they relate to application in the NPP industry for legacy assets and SSCs**

As indicated by the evolution of the levels, the first level is a simplified digital representation that is linked to an asset management tool while the fifth level include highly advanced DTs with capability of preventative and predictive maintenance with autonomous decision-making. The descriptions for these levels were defined based on collaboration with NPP operators and engineers that provided insight of current management trends and procedures. As 3D scanning technology becomes more prevalent in the industry, the idea of obtaining 3D point clouds of a large-scale facility has become a standard procedure in creating digital records. Therefore, it was determined that a 3D point cloud with auto-linking compatibility would be the basic level of DTs to provide the fundamental virtual representation without the investment of modelling or additional features. With each additional level, the DTs become more complex with modelling functionalities and feedback capabilities. The utmost level is Level 5, which includes a full geometric and physics model,

autonomous decision-making for performance, and auto-linking compatibilities to interact with related DTs.

The level of DTs for a specific asset should be selected based on a variety of criteria and roles. As indicated by the definitions, each level is intended to serve a slightly different purpose in terms of asset management. The following section will provide a detailed explanation for choosing the appropriate level and what support documents should be used as reference in the decision-making process.

#### **4.4.1 Supporting Documents for Defining DT Level**

The determination of an appropriate DT level for a given asset will be made by the operator or engineer who is implementing the DT. This responsible person should be supported and consult with a multi-disciplinary team to account for NPP system complexity, discipline specific SSC knowledge, and communications requirements for institutional knowledge management. The main input factors for determining an appropriate level of DT include the asset support documents, a preliminary investigation of the SSC, and the level of criticality or safety rating of the SSC. The following descriptions of these contributors should act as a support tool when determining the level of DTs.

**Support Documents:** Support documents include mechanical or construction documents, operating and maintenance manuals or procedures, existing 2D imagery, 3D point data, existing models or simulations of the asset, and other documentation that relate to the geometric and operational properties of the asset to be modelled. A higher quantity and quality of support documentation could provide more detailed information thus should indicate that higher-level DTs could and perhaps should be achieved. Minimal or incomplete support documentation indicates lower-level DTs could be achieved or that more data needs to be acquired, if a higher-level DT is desired.

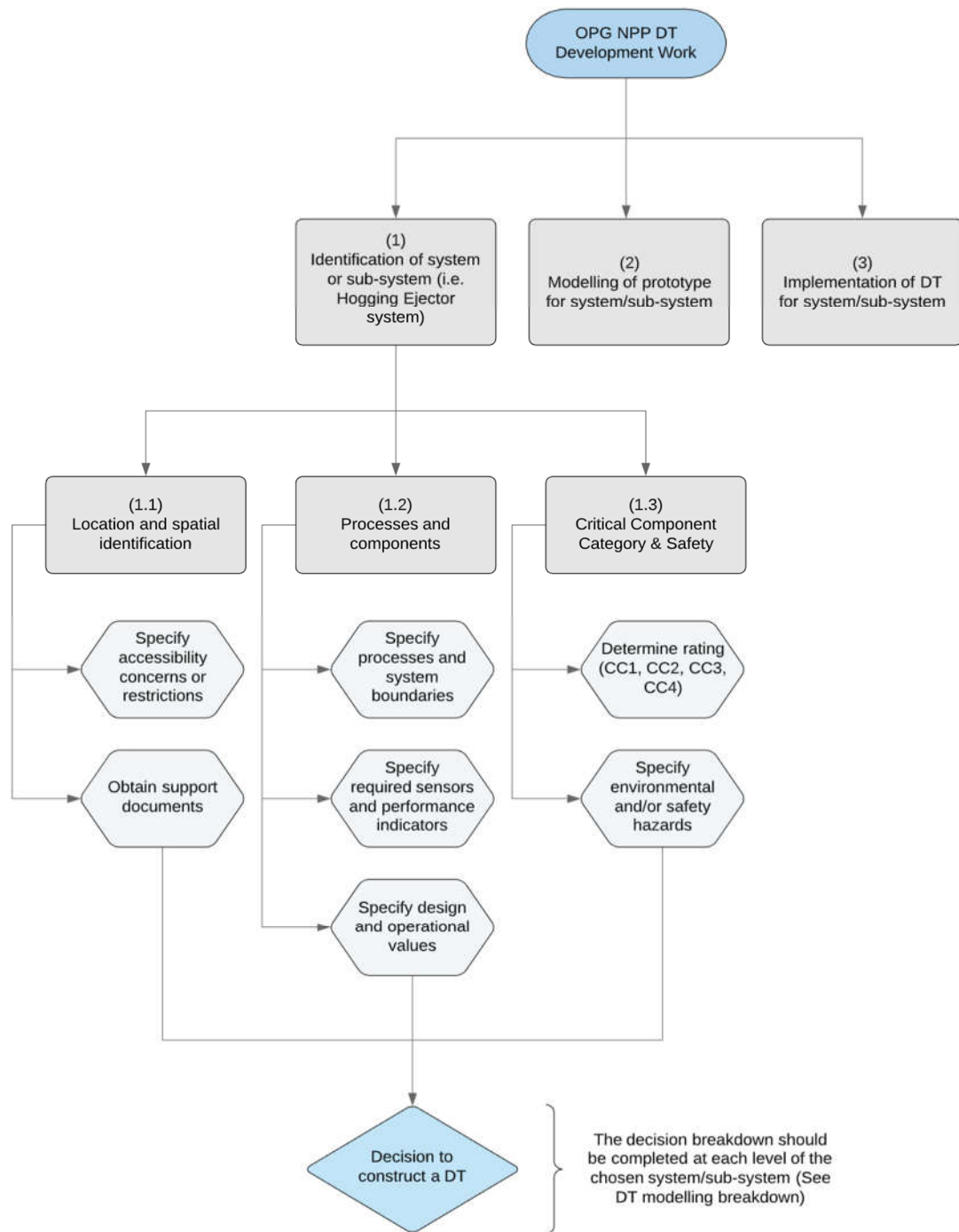
**Preliminary Investigation:** A preliminary investigation of the asset of interest should be conducted to identify past events and potential failure mechanisms. This will disclose areas of concern related to the operation of the asset and present key behaviours that should be modelled or simulated. The investigation should also cover accessibility restrictions, potential hazards, and related systems or sub-systems that could be affected. This step is extremely useful as it prioritizes the development of DTs for SSCs that have exhibited issues causing plant outages or shutdowns. A system with reoccurring issues or severe consequences from failure should be developed at a higher level, whereas less significant assets could be modelled with a lower level DT.

**Criticality and Safety:** The criticality of legacy assets was previously described when defining SSCs in NPPs. The method of rating an asset’s criticality may vary depending on the organization and their official procedures. Despite variance in procedures, the underlying operating principle is to treat highly critical assets with higher level DTs and less critical and replaceable assets with lower level DTs. A higher level DT will provide more detailed feedback regarding the asset’s current and future operations, thus indicating potential failures that could have a high impact on overall plant operation and safety.

The definitions above are subjective according to organization and engineering judgment. Depending on the NPP FM procedures and departments, the DTs level may also be dependent on the role of the operator using or implementing DTs. For example, levels 1 through 3 would be appropriate for departments that are interested in knowing recent updates to an SSC, locating and visualizing an SSC, or using the DTs to gather measurements and other visual parameters. Meanwhile, DTs that are level 4 or 5 would be more practical for departments involved in predictive and preventative maintenance. These levels would allow operators to model potential scenarios to better understand when an asset requires maintenance and to provide an updated lifespan of the SSC.

#### **4.5 Proposed Task Breakdown Structure for DTs**

Defining the DTs level for a given asset is one of the many steps involved in the implementation of DTs. To further guide the decision-making process, a task breakdown structure for DTs is proposed in Figure 4.3. This task breakdown structure identifies the primary tasks related to DTs, which relate to the support documents in the previous section.



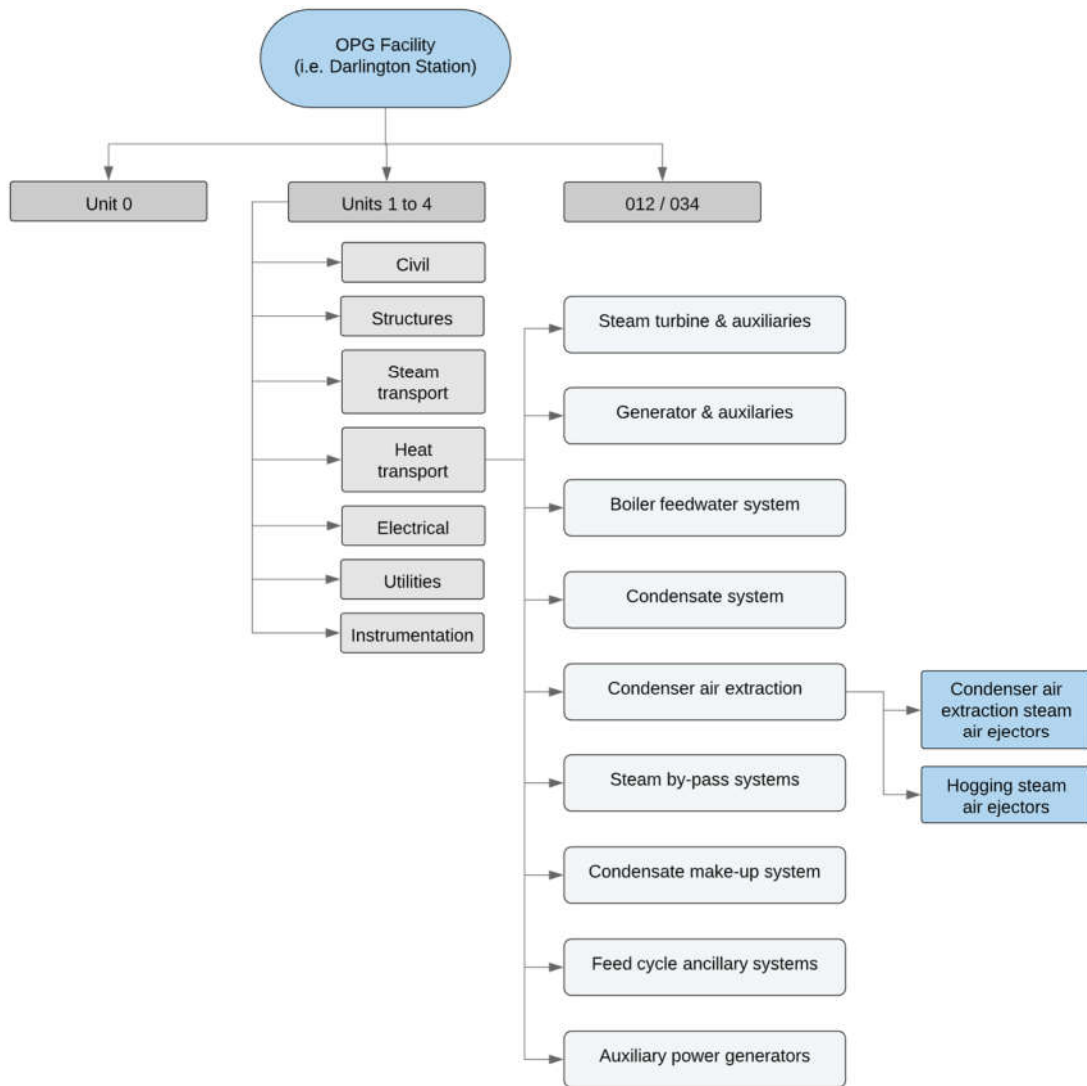
**Figure 4.3 This task breakdown structure is a support tool that identifies the tasks that should be completed for decision making related to implementation of DTs for a specific SSCs**



The first level of this task breakdown structure identifies the primary tasks of implementing DTs: 1) identification of SSC, 2) modelling of prototype for SSC, and 3) implementation of DT. The framework and concepts presented in this thesis relate to the branches that follow the identification of SSC. As shown in the figure, after identifying the SSC the necessary tasks relate to the supporting documents that were described in the earlier section of this chapter. This diagram simplifies factors that should be considered for each supporting document.

#### **4.6 Using the Framework and Support Tools**

The framework and tools provided in this chapter are intended to act as support when implementing DTs for any legacy asset specific to NPP FM. Therefore, when implementing the framework there is room for interpretation depending on the asset of interest and intended use of the DTs. When considering implementing DTs, it is necessary to gain a sense of the systems and sub-systems that relate to the SSC of interest. This could be completed by creating a system map related to the SSC in question. An example of a systems map is shown in Figure 4.4. The system map is intended to show the network of SSCs or legacy assets and how they are connected within the facility. This type of map is facility and organization dependent, but the concept of connecting SSCs to understand the overall network is important for the functionality of dependent or interdependent DTs.



**Figure 4.4 Example of a branch of a systems map to understand the related systems and sub-systems of an SSC**

Mapping the related systems and sub-systems identifies where DTs could be constructed and linked with the DT of the SSC of interest. It is important to note that the levels of DTs for related systems and sub-systems may vary and the process of determining an appropriate level should be carried out for each SSC individually. The connections between SSCs may be taken into consideration when implementing DTs, however it is not necessary to use the same level for each DT related to a certain system or sub-system.

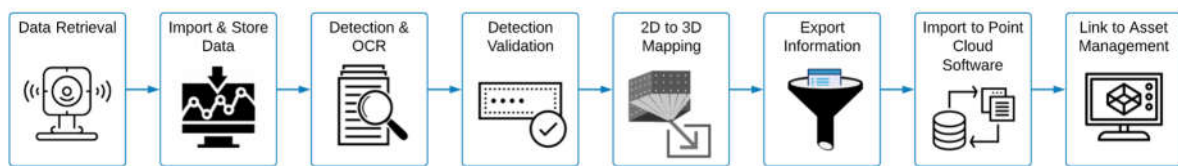
## Chapter 5

### Auto-linking for a Digital Database of Legacy Assets

The Auto-linking algorithm was developed in the process of improving the efficiency of asset management for NPPs. In relation to the layered framework, auto-linking may be considered an approach to the computation layer for creating a digital database of assets. This chapter includes the definition of auto-linking, a description of its applications and uses, the methodology to this approach, and a detailed discussion regarding its development.

#### 5.1 Definition of Auto-linking and Description of Applications

The overlying need for the layered framework is a more efficient approach to implementing DTs for legacy assets in NPP. Automating the process of detecting assets and components and the creation of a digital database that relates to asset management will accelerate the management-related tasks. This sub-section will cover the definition of Auto-linking and its various applications. Later sub-sections detail the methodology and technology required to achieve the intended outcome. Auto-linking is defined as the automatic detection and linking of assets from the original scan data to the final asset management tool. Between the detection of the asset and the linking to the asset management tool, the algorithm also recognizes equipment information, determines its relative location and stores the identifying information to a data format that can be imported to the point cloud software. The pipeline<sup>1</sup> for the Auto-linking algorithm is shown in Figure 5.1.



**Figure 5.1 Auto-linking pipeline**

Each step of the Auto-linking pipeline is described below. This pipeline was developed for the specific use case of NPP facility management with the potential for adoption for other applications.

---

<sup>1</sup> A pipeline is a sequence of steps or processes that when combined yield the output of interest. Certain elements of the pipeline may be executed in an iterative manner to accomplish the model objectives.

**Data Retrieval:** The auto-linking pipeline begins with retrieving image and 3D scan data related to the area of interest. There are several 3D scanning methods available to gather sufficient data to be used for this process.

**Import & Store Data:** It is necessary to store the gathered data in a folder with a unique identifier so the correct data can be called in the following steps.

**Detection & OCR:** Code was developed in Python that runs a trained object detection algorithm on the data, then processes the image, and runs OCR on the processed image to detect the equipment ID information. The script is described in detail in later sections and presented as part of Appendix B.

**Detection Validation:** This validation applies to the information extracted from the previous step. The information that was extracted from the previous step can be validated by comparing the equipment ID to an existing database of the expected components within the scan area.

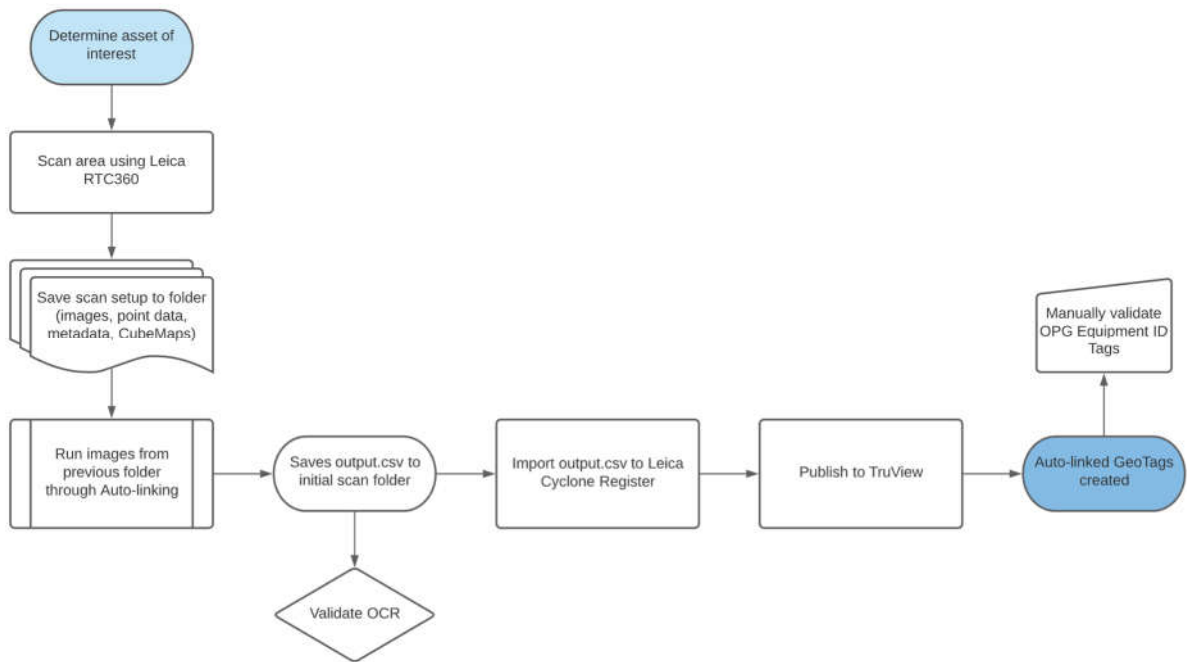
**2D to 3D Mapping:** This step is necessary for determining where the detected object is located within the 3D point cloud. The 2D to 3D mapping is commonly done through backend software processes when overlaying HDR imaging with 3D point data. The transformation to complete this step is dependent on the scanning and imaging software used and their intrinsic properties.

**Export Information:** Once the detection is complete, the equipment ID has been read and validated and the 3D coordinates of the object are obtained, the information can be exported in the appropriate format. This should be saved to the initial folder from the second step of the pipeline.

**Import to Point Cloud Software:** The software used in this step is dependent on the laser scanner technology and related software. The software will determine what format should be used for import.

**Link to Asset Management:** Once the pertinent information has been imported into the point cloud software, the point cloud can be published to the asset management tool that is used. The operator now has a tag within the 3D point cloud that contains information corresponding to a specific asset.

The work breakdown for Auto-linking is shown in Figure 5.2 as it would apply to OPG procedures. This workflow identifies the primary tasks that are required to achieve the complete pipeline.

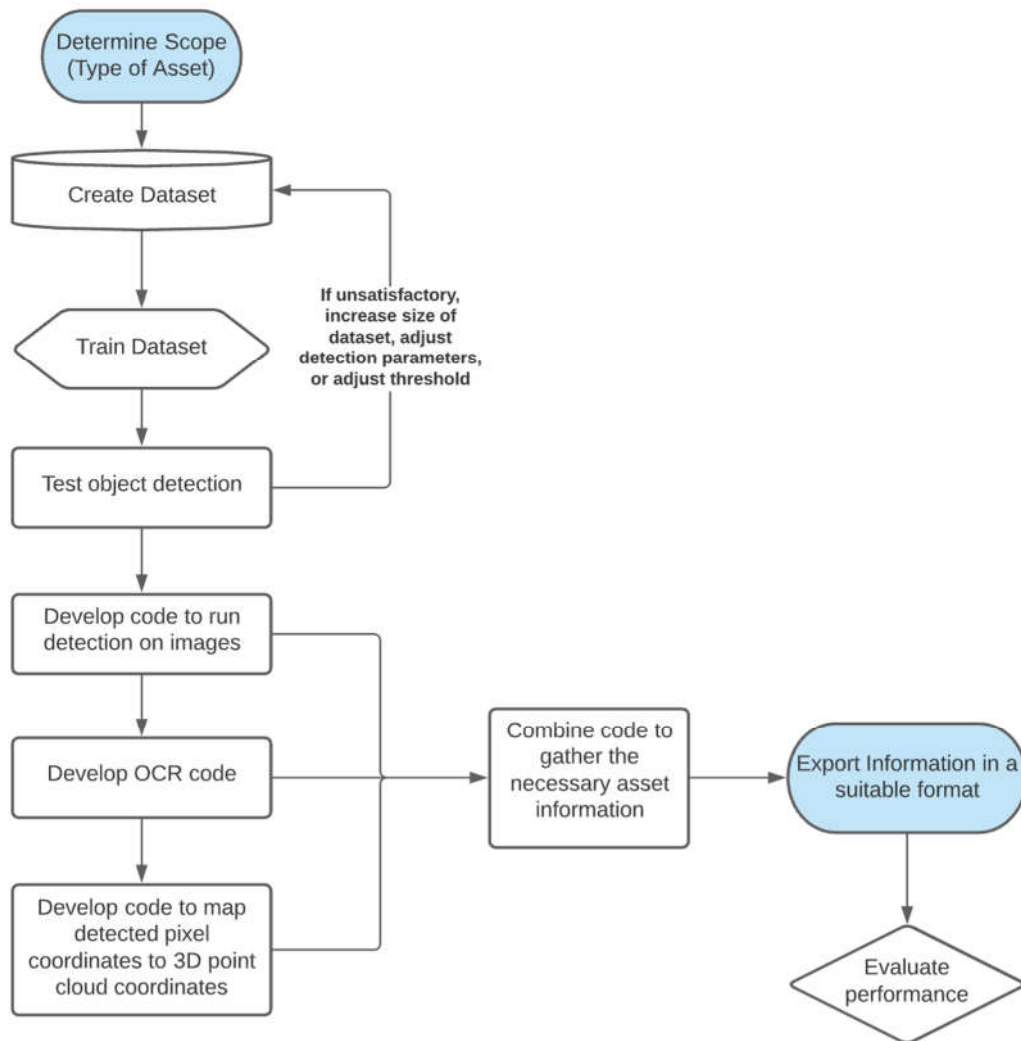


**Figure 5.2 Workflow for Auto-linking in OPG Asset Management Procedure**

The Auto-linking pipeline can be applied to facilities with legacy assets to automate the process of digitally documenting and monitoring assets. Automating the previously manual process of searching point clouds for equipment reduces the time spent on this task and increases the overall productivity of the asset management procedures. This automated process partly makes up the Level 1 DTs since it provides virtual representation of the physical environment and has linked assets within the asset management tool. It provides basic feedback information such as equipment identification, related documents, or files, and an updated visual of the asset.

## 5.2 Requirements and Methodology

The requirements and workflow of Auto-linking are described in detail below. This section also includes a description of technology and software that will complete the processes described in the previous sub-section. Figure 5.3 details the workflow of creating the Auto-linking algorithm.



**Figure 5.3 Auto-linking pipeline developed for automation of digitizing the SSC database and asset management procedure**

The intent in developing this workflow is to clarify the order of tasks that contribute to the Auto-linking algorithm. The background information and current trends related to topics such as Object Detection, OCR, and 2D to 3D mapping are included in Chapter 2. The following software and tools are required to complete the process:

- OpenCV: an open source library primarily used for computer vision and machine learning applications. The library contains over 2500 optimized algorithms that can be used in detection, recognition, segmentation and other projects. (OpenCV Team, 2021)

- CUDA Toolkit for GPU computation: a NVIDIA product that creates an environment that allows for GPU-accelerated applications such as machine learning and computer vision. (NVIDIA, 2019)
- Darknet: an open source neural network framework (Redmon, 2016)
- Yolo\_label.exe: a package that functions as an image-labelling tool for object detection. (Yonghye Kwon, 2019)
- Python and related packages: programming language for software development and other applications. (Python Software Foundation, 2021)

Detailed descriptions of how these tools contribute to the Auto-linking performance are included in Chapter 6.

### **5.2.1 Object Detection**

The general definition of object detection includes the identification and recognition of objects based on their features in images and videos. The concept has been studied and advanced over the past two decades and has multiple applications, including face recognition, image or video segmentation, and object detection for autonomous vehicles, to name a few. Popular approaches to object detection derive from machine learning and deep learning approaches, both of which were explored and tested for the Auto-linking application. The processes of applying these approaches are described in detail in the following sections.

The purpose of employing object detection in this thesis is to minimize the manual labor of identifying assets within a point cloud by first detecting and labelling equipment tags in raw image data prior to importing to the point cloud. As such, object detection was explored as a solution to automating this process.

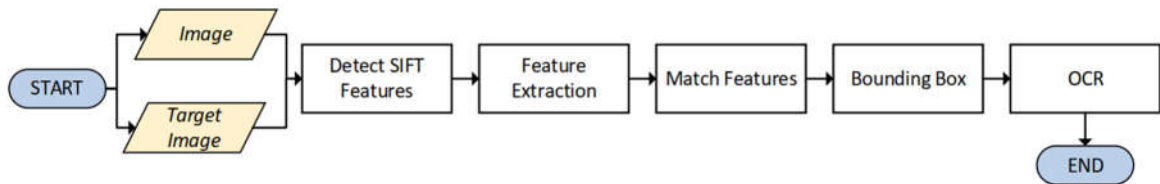
### **5.2.2 Machine Learning-Based Approaches**

Machine learning approaches that have been commonly employed for object detection include Viola-Jones object detection framework, Scale-invariant feature transform (SIFT), and histogram of oriented gradients (HOG) features. Along with these popular approaches, fundamental computer vision techniques were researched as approaches. As such, during the process of determining an appropriate machine learning-based approach for the object detection of equipment ID tags, the following approaches using machine learning concepts were examined:

- SIFT approach
- MSER approach
- Hough Transform approach
- White Regions detection approach

These approaches were developed using MATLAB and evaluated based on a subject area (“test bed”) in the University of Waterloo Sensing Lab. The subject area was set up with multiple sample equipment ID tags and scanned using a FARO Focus Laser Scanner. Below is a detailed description of each approach and their evaluations. The MATLAB scripts associated with the machine learning-based approaches are included in Appendix A. Also included in Appendix A are the detailed results from each approach.

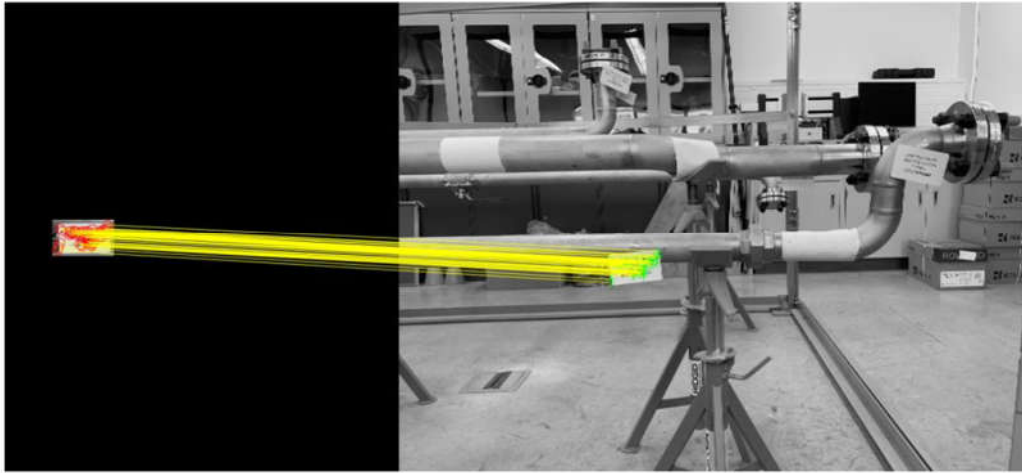
The SIFT approach enforced MATLAB’s SURF features to match feature points between the target and an overall scene image. This approach is invariant to scale, rotation, and illumination; hence, it was thought to be an advantageous method to detect features in a range of environments and orientations. The process for this approach is shown in Figure 5.4.



**Figure 5.4 Pipeline for SIFT approach demonstrating the matching of features between a target image and a scene image**

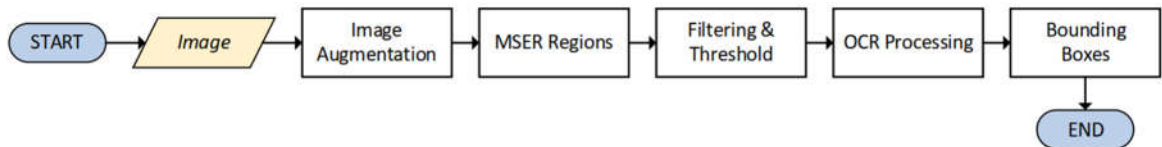
Figure 5.5 shows an example of how the SIFT approach detects an object within a scene using SIFT features. The SIFT features are mapped from the target image to the object in the scene image.





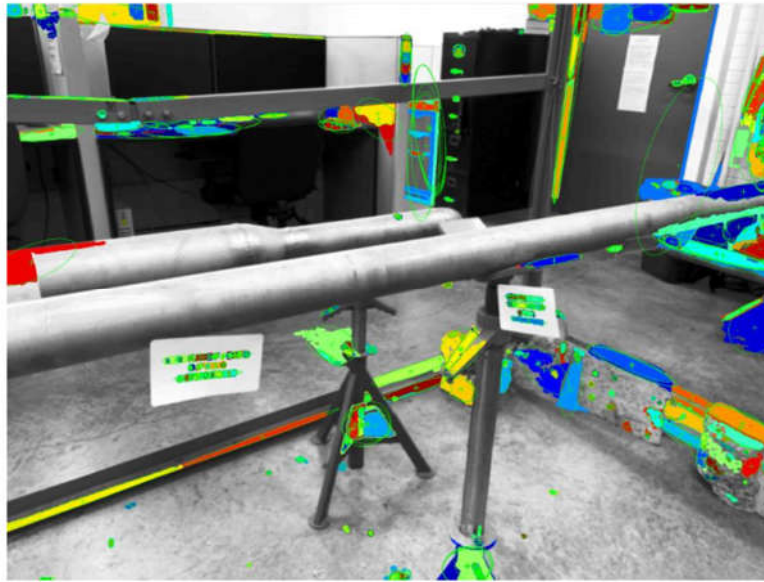
**Figure 5.5 Example of a tag that was detected using SIFT approach**

The MSER approach used Maximally Stable Extremal Regions (MSER) to detect patterns that could indicate an equipment ID tag. The rationale behind this approach was the consideration that the equipment tags are consistently white and black and may contrast with their environment thus producing consistent patterns from the MSER features. The detection of these regions would indicate the possible presence of an equipment ID tag. The OCR was intended to validate if the region contained an equipment ID tag. The pipeline for this approach is shown in Figure 5.6.



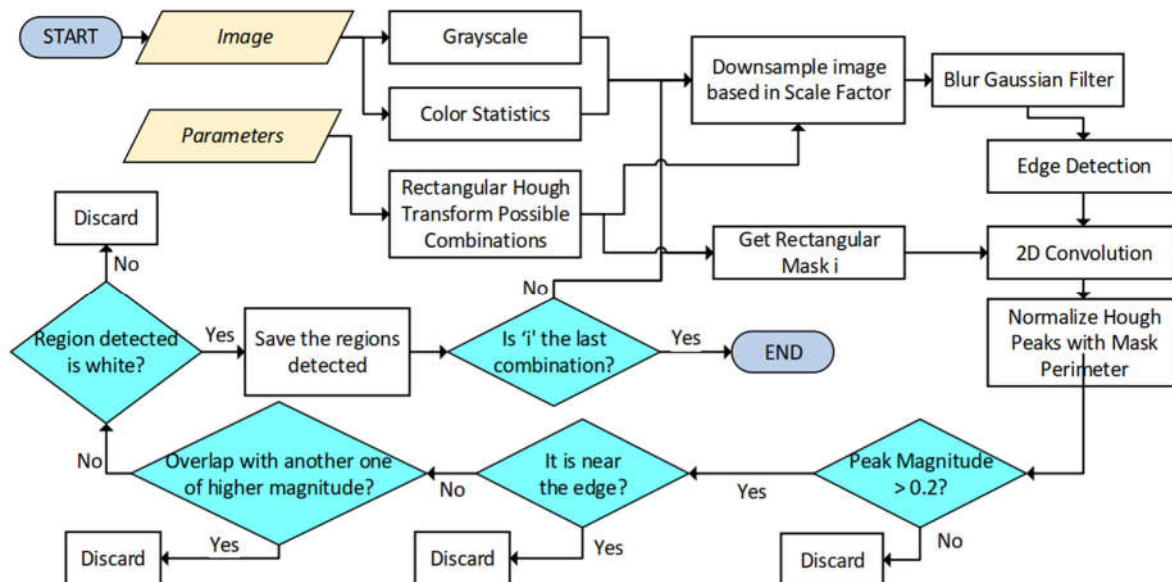
**Figure 5.6 The pipeline for the MSER approach to show how the equipment ID tag is detected using MSER features**

Below in Figure 5.7 is an example of the MSER features that were detected in the scene image. After OCR processing it was expected to be able to detect the equipment ID tag with bounding boxes.



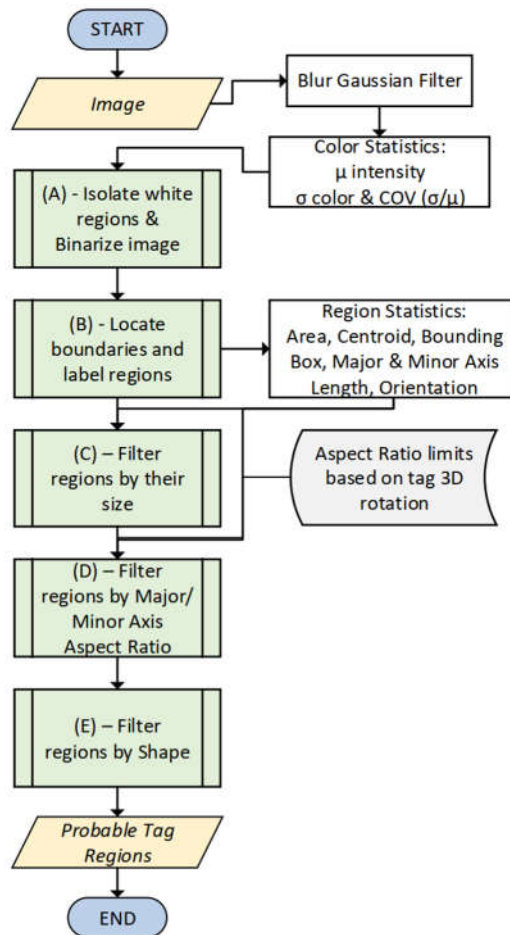
**Figure 5.7 MSER regions detected in an overall scene to be used for tag detection**

The third approach followed the principles of the Hough Transform which is a technique often used for line detection. The Hough Transform performs a convolution of lines at different positions and angles over the image. This approach used the kernel of a projection of an equipment ID tag rotated along three different axes of rotation. The pipeline for this approach is shown in Figure 5.8.



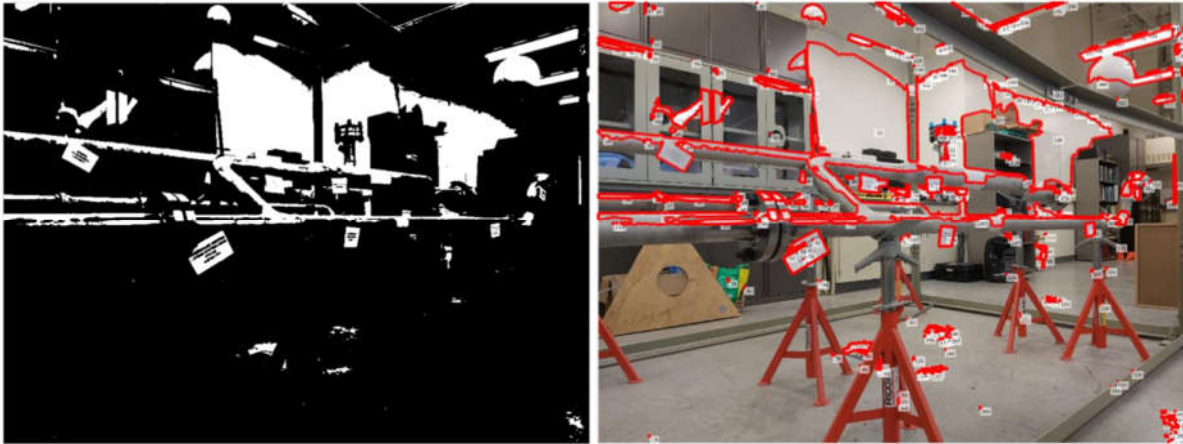
**Figure 5.8 Pipeline to demonstrate the process of using the Hough Transform for equipment ID tag detection**

The final approach was called the White Regions approach, which was developed because of the format of the equipment ID tag. This approach aimed to isolate white regions within the image which would indicate the possible location of an equipment ID tag. This approach sets a baseline of colour and brightness then discards regions of an image that do not meet the baseline requirements. The pipeline for this approach is shown in Figure 5.9.



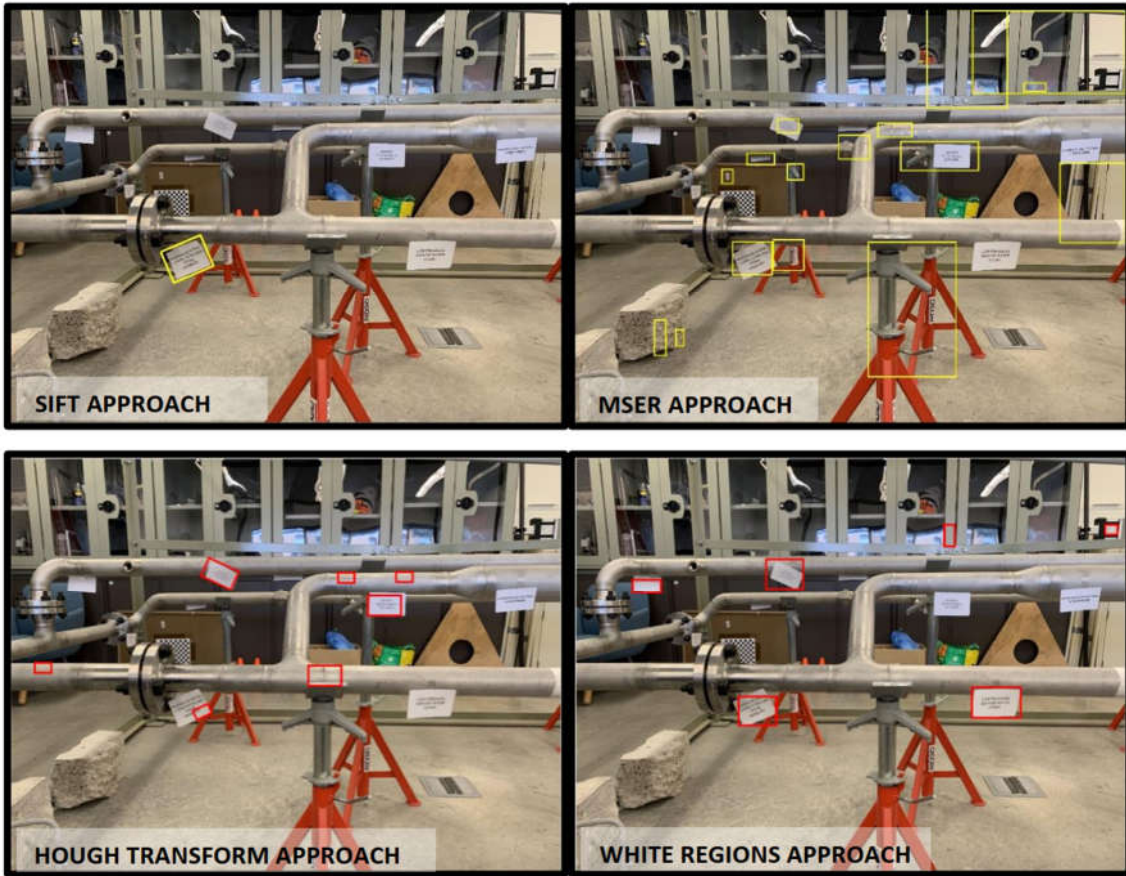
**Figure 5.9 The pipeline related to the White Regions approach which processes images and sorts them based on a set of baseline parameters to detection potential equipment ID tags**

Figure 5.10 demonstrates how the detection of white regions in an image could determine the location of equipment ID tags.



**Figure 5.10 Example of the white regions approach which identifies white regions within the scene images then bounds potential equipment ID tags**

Each approach was evaluated based on the test setup in the UW Sensing Lab. The test setup included 18 sample tags. Of these sample tags, 10 tags were considered “Target Tags” with correct format and text, while 8 tags were “Inconsistent” based on their format and text. The objective of the testing was to determine how well each approach was able to identify the Target Tags and discard Inconsistent Tags. Examples of each approach are shown in Figure 5.11 which evaluates the same scene under each approach.



**Figure 5.11** Examples of each approach using the same scene image

The results of this experimental test setup are included in Appendix A. Furthermore, Table 5.1 The advantages and disadvantages of each machine learning-based approach based on the experimental test setup.

**Table 5.1** The advantages and disadvantages of each machine learning-based approach based on the experimental test setup

Approach	Advantages	Disadvantages
SIFT Approach	<ul style="list-style-type: none"> <li>• Detects features that vary in scale, rotation, and illumination</li> </ul>	<ul style="list-style-type: none"> <li>• Requires a target image of an object that exists within the scene image therefore would require additional pre-processing of data</li> </ul>

	<ul style="list-style-type: none"> <li>• Worked well in scenes with similar equipment ID tag formats</li> </ul>	<ul style="list-style-type: none"> <li>• Computational exhaustive and would require a significant amount of time to complete the detection of multiple scenes</li> </ul>
MSER Approach	<ul style="list-style-type: none"> <li>• Did not require trained data set or input images</li> <li>• Accurate detections that included the text of the equipment ID tags for OCR reading</li> </ul>	<ul style="list-style-type: none"> <li>• Results consistently included false negatives therefore would require a high amount of post-processing to sort through the detections</li> <li>• There were inaccurate results in scenes with large amounts of equipment ID tags</li> </ul>
Hough Transform Approach	<ul style="list-style-type: none"> <li>• Uses classic computer vision algorithms which do not require a large image set for training</li> <li>• Higher accuracy compared to the first two approaches</li> </ul>	<ul style="list-style-type: none"> <li>• Computational expensive as it requires a large amount of memory and time for processing each image making it impractical for this application</li> </ul>
White Regions Approach	<ul style="list-style-type: none"> <li>• Uses classic computer vision algorithms and does not require a large data set or samples for calibration</li> <li>• Regions are well defined making it simpler to validate the detections</li> <li>• Implementation and processing are fast compared to other approaches</li> </ul>	<ul style="list-style-type: none"> <li>• Tags with low contrast against the scene environment are not detected or the detection includes the background noise</li> <li>• Is only capable of detecting rectangular shapes</li> <li>• It discards tags that are detected with other small surfaces</li> </ul>

Overall, the results from these machine learning-based approaches were unsatisfactory because of a multitude of disadvantages and overall inadequate precision and recall performance. Primarily, it was

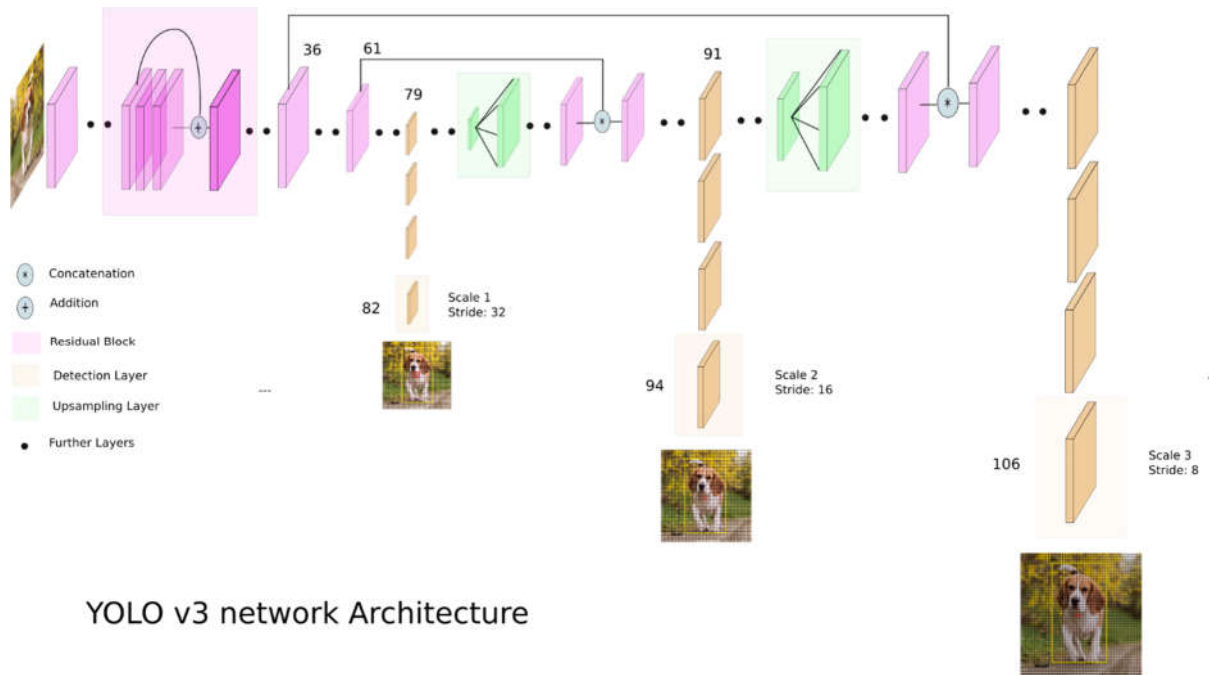


found that each approach required custom filters or specified-baseline-targets to conduct the image processing. This amount of pre-processing and customization makes the approaches unreasonable for smooth and practical implementation for large-scale projects such as for NPP facilities, which contain a high amount of legacy assets. Although these approaches would be appropriate for small datasets or individual detection scenarios, it was determined that another approach should be explored for the large datasets this project involves. The primary finding is that the selected approach should require minimal manual pre-processing of images and the detection should be independent of the environment.

### **5.2.3 Deep Learning-Based Approaches**

Based on the unsatisfactory performance of the machine learning-based approaches previously discussed, a deep learning approach was explored in further detail. Common deep learning approaches for object detection include Regional-based Convolutional Neural Networks (R-CNNs), You Only Look Once (YOLO) model, Single Shot MultiBox Detector (SSD), Feature Pyramid Networks for Object Detection (FPN), RetinaNet, EfficientNet and others. The approach taken in this thesis employed the YOLOv3 model with Darknet-53 for training and detection.

YOLOv3 is an object detection model that is configured using a combination of Residual Network (ResNet) and Feature Pyramid Networks. The model uses a multi-label classification method with binary cross-entropy loss for class predictions during training. The prediction is executed across three different scales which are down sampled according to dimension sizes 32, 16, and 8. The architecture for YOLOv3 is shown in Figure 5.12 (Kathuria, 2018).



**Figure 5.12 The YOLOv3 network architecture (Kathuria, 2018)**

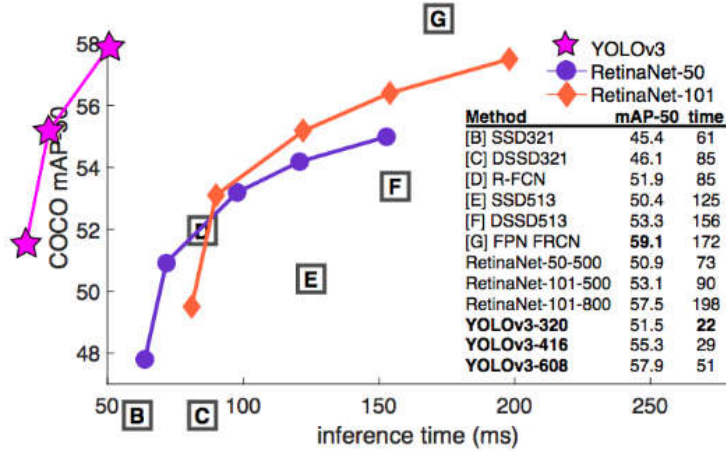
The feature extractor used in YOLOv3 is called Darknet-53 based on the 53 convolutional layers that are involved, as shown in Figure 5.13 (Redmon & Farhadi, 2018). The original version of the YOLO model used a Darknet version with only 19 layers. The concept of skip connections to propagate through deep layers was developed by ResNet and incorporated in this model to extend the network from the original 19 layers to 53 layers. This evolution improved the detections and efficiency of the overall network.



	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1x	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2x	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8x	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8x	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4x	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
Connected		1000		
Softmax				

**Figure 5.13 The feature extractor known as Darknet-53 that is used in YOLOv3 (Redmon & Farhadi, 2018)**

YOLOv3 has been evaluated as a fast and effective object detector compared to its competitors such as Faster R-CNN and RetinaNet, as shown in Figure 5.14 and Figure 5.15 (Redmon & Farhadi, 2018). It was proven to be faster than SSD variants and not quite as powerful but comparable to RetinaNet using COCOs average mean AP metric.



**Figure 5.14 Performance of YOLOv3 against other algorithms based on a common metric: COCO mAP-30 and inference time (Redmon & Farhadi, 2018)**

	backbone	AP	AP <sub>50</sub>	AP <sub>75</sub>	AP <sub>S</sub>	AP <sub>M</sub>	AP <sub>L</sub>
<i>Two-stage methods</i>							
Faster R-CNN+++ [5]	ResNet-101-C4	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN w FPN [8]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN by G-RMI [6]	Inception-ResNet-v2 [21]	34.7	55.5	36.7	13.5	38.1	52.0
Faster R-CNN w TDM [20]	Inception-ResNet-v2-TDM	36.8	57.7	39.2	16.2	39.8	<b>52.1</b>
<i>One-stage methods</i>							
YOLOv2 [15]	DarkNet-19 [15]	21.6	44.0	19.2	5.0	22.4	35.5
SSD513 [11, 3]	ResNet-101-SSD	31.2	50.4	33.3	10.2	34.5	49.8
DSSD513 [3]	ResNet-101-DSSD	33.2	53.3	35.2	13.0	35.4	51.1
RetinaNet [9]	ResNet-101-FPN	39.1	59.1	42.3	21.8	42.7	50.2
RetinaNet [9]	ResNeXt-101-FPN	<b>40.8</b>	<b>61.1</b>	<b>44.1</b>	<b>24.1</b>	<b>44.2</b>	51.2
YOLOv3 608 × 608	Darknet-53	33.0	57.9	34.4	18.3	35.4	41.9

**Figure 5.15 Metrics that measure the quality of the YOLOv3 algorithm (Redmon & Farhadi, 2018)**

The few limitations of YOLOv3 include imperfect alignment of the bounding box with the actual object and lower performance with medium to large objects. Since the scope of this project included relatively smaller objects and did not require precise alignment of the detection, these limitations did not impede the decision to use the YOLOv3 model.

After deciding to take the deep learning-based approach for the equipment ID tag detection, the following step was to implement by training and testing. Since the scope of this project involved custom equipment ID tags, it was necessary to create a custom dataset to train the YOLOv3 algorithm. Approximately 5,000 images were provided by OPG for training purposes. An image-labeling tool, Yolo\_label.exe was used to label the images for the training dataset. To use this tool, a

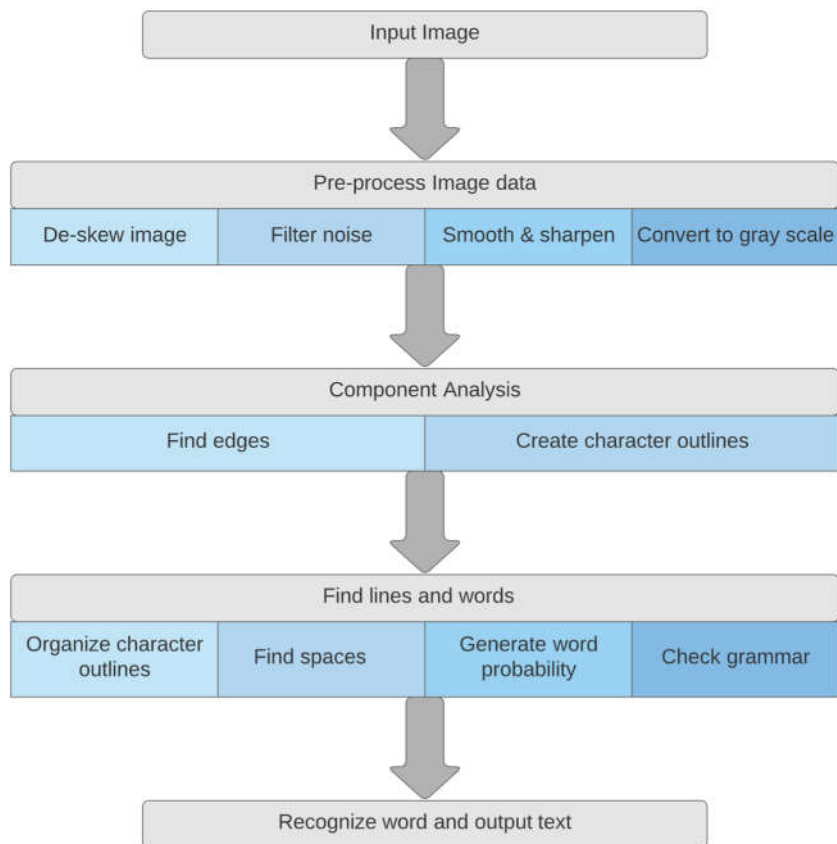
text document was first created with a list of the image classes. These image classes are determined based on the user and were used to label the objects in the image dataset. For this project, three classes were used based on the appearance of common equipment ID tags: OPG Equipment Tag A, OPG Equipment Tag B, and OPG Equipment Tag C.

The training and testing of this custom dataset were completed in two stages. The first stage was conducted on a small dataset of less than 500 images. The second training and testing was conducted on a larger dataset of 1500 images. The intent of conducting two stages was to first determine the feasibility of the deep learning approach without using exhaustive computer processing. The second stage that involved a much larger dataset was employed to improve the results of the detector. The higher volume of training images leads to improved results, because the algorithm is trained on additional orientations, environments, and conditions.

Following the training and testing of the custom dataset, a Python script was written to call a set of testing images from a folder and run the YOLOv3 object detection using the final weights from the custom training. The code also stores the pixel coordinates from the detected bounding box. The pixel coordinates are first used to crop the image to pre-process for OCR, then they are used to map the detection into 3D point coordinates such that they can be transferred into the 3D point cloud. The full code is presented in Appendix B.

#### **5.2.4 Implementing OCR to Extract Equipment ID**

OCR is the recognition and conversion of text in images to machine-encoded text. Applying OCR to the detected objects is a necessary step for identifying that the detected object is an equipment ID tag. The results from the OCR will also be used for exporting essential information at the end of the Auto-linking process. The process for OCR is as follows (Figure 5.16):



**Figure 5.16 Breakdown structure of OCR**

The breakdown above describes the general process implemented by typical OCR packages. Several OCR packages were tested on a limited dataset to determine which package would be appropriate for this application. Based on this preliminary testing of the packages, keras-ocr (Morales, 2019) was chosen. Once the OCR tool was chosen, it was written in the code to run on the cropped image from the detection algorithm. Keras-ocr detects all possible text, words, phrases that might be within the image. To filter through these potential results, the following filters were applied:

1. Text that contains numbers
2. The largest string is most likely an Equipment ID tag

By applying these filters the output text became more accurate and precise. In addition to the OCR, a text recommender was used to auto-correct the OCR results such that they more closely match the Equipment ID tag format. Further explanations including how the OCR was implemented and the methodology behind the text recommender are described in Chapter 6.

### 5.2.5 Mapping from Pixel Coordinates to 3D Point Cloud Coordinates

An essential step in completing the Auto-linking process is mapping the detected object to its corresponding 3D point cloud location. This process can be completed using an algorithm that transforms 2D pixel coordinates to 3D point coordinates. This algorithm is included in the back-end development of scan registration and display software to overlay images with point data.

In the development of the Auto-linking algorithm, the Leica RTC360 3D Laser Scanner and Leica Geosystems software were considered. This laser scanner captures 3D point data and HDR imagery which can be directly published to their software known as Leica Cyclone Register 360. After publishing to Cyclone Register the Leica TruView software can be used for viewing, sharing, and marking up scans.

The method of mapping from pixel coordinates to point coordinates used in this research required the .JPG image of the scene, a matching 16-bit .PNG image, and an .XML file that contains the scale and offset values. The .PNG and .XML files were provided by Leica Geosystems for this project, however the SDK version of the software is required to have access to the mapping function. Once these files were provided, the world coordinates could be computed as follows:

1. Determine .PNG row and column values based on detection bounding box coordinates (rgb).

$$row = \frac{x_{detection}}{Width_{rgb}} * Width_{png}$$

$$col = \frac{y_{detection}}{Height_{rgb}} * Height_{png}$$

2. The .PNG is a 16-bit image with depth values. The above row and column are used to extract the target point integer.

$$[point_{integer}]_{3x3} = [PNG_{matrix}](row)(col)$$

3. Once the point integer coordinates are determined, the scale and x, y, z offsets from the .XML can be applied to compute the world coordinates.

$$[point_{world}]_{3x3} = scale * [point_{integer}]_{3x3} + [x_{offset}, y_{offset}, z_{offset}]$$

This last step determines the coordinates of the detected object in 3D space such that it can be located within the point cloud.

### **5.2.6 Linking Detections to Asset Management Database**

The final step of the Auto-linking process is to import the essential information that is used to create a tag in the 3D point cloud software. This involves importing the data that was obtained from the previous steps into Leica Cyclone Register. Next, the files would be uploaded to Leica TruView which would display the equipment ID tags that are detected. Once this final step has been completed, operators or engineers can use TruView functions to edit information related to the GeoTag, if necessary.

### **5.2.7 Validation of Database**

Given the importance of asset naming in a facility such as NPPs, it is essential to validate the information extracted from the Auto-linking pipeline in creation of a digital database. There are two options for validation of the database, as described below.

**Option 1: Search and Match.** This technique would require a list of existing assets within the area of scanning. A list will be created from the information extracted from the Python script including the equipment ID. Next, this unique identifier will be searched for against the list of existing equipment to determine if the equipment ID had been correctly read using the OCR. This can be completed using a text recommender or autocorrect that compares the OCR text to possible text and selects the most correct text based on the Jaccard distance (Wang, Lu, Wang, Liu, & Zhou, 2017).

**Option 2: Human operator.** Another option for validating the digital database is to have an operator confirm the unique identifier by reading the detected equipment ID tag in the point cloud where it was automatically tagged. While this may be more time consuming, it provides a higher level of validation and still reduces the time spent on manually tagging equipment in the point cloud.

## **Chapter 6**

### **Experimental Analysis of Auto-linking**

An experimental analysis was carried out to demonstrate the applicability and efficiency of the Auto-linking process. The experimental analysis included both training and testing data from the OPG Darlington Facility. The scan data provided was obtained from their Training Facility to comply with security regulations. The following sub-sections describe the experimental design, assumptions and limitations, methodology, software, and an analysis of the results.

#### **6.1 Experimental Design**

The experimental analysis was designed to demonstrate how Auto-linking can improve the productivity and efficiency of labelling assets for the purpose of asset management. The objectives of this experimental design include:

- Evaluate the object detection using the YOLOv3 algorithm
- Evaluate the OCR and data output from the Auto-linking code
- Demonstrate the mapping of the detected image coordinates to point coordinates

This section describes the current procedure of tagging assets as practiced by OPG then details the scope of the experimental design, evaluation metrics, and how Auto-linking addresses the challenges of the existing procedures.

##### **6.1.1 Current Procedure for Tagging Assets**

The current procedure implemented for tagging assets at OPG is known as “geo-tagging”. Geo-tagging creates a tag in the 3D point cloud that contains useful asset management information about the corresponding SSC, including equipment ID, SSC type, notes, and attachments such as links, documents, or images. This allows OPG employees, operators, and engineers to gather information about an SSC directly from the TruView environment which includes a point cloud with HDR imaging overlay.

The primary software used by OPG includes Leica RTC360 Scanner with Leica Cyclone for scanning and registering, and TruView Enterprise (TruView) to present and share the 3D point cloud. TruView is a Leica Geosystems Product that has various functionalities that are useful for facility management including sharing point cloud data, virtual tours around the scanned area, tools such as

zoom, measure, and markup, and GeoTags and Hyperlinks. After registering a scan using Leica Cyclone, the scan data can be published directly to TruView for OPG wide applications.

After publishing the scan data to TruView, the point cloud is accessible on the OPG network for multi-purpose uses across departments. An operator or engineer will be directed to the TruView Application and search for the NPP site of interest. Once a site is selected, the operator will choose one of the following tabs: SiteMap, Setups, GeoTags, Snapshots, Assets, or Info. The content of each tab is described below in Table 6.1.

**Table 6.1 Description of the tab options that are used in the asset management tool TruView**

Tab Identifier	Description
Site Map	This is a 2D map of the site, it contains icons to indicate scanner setup locations. Click on a setup icon to load the corresponding 3D point cloud.
Setups	These are the scanner locations that were used to generate the 3D point cloud. Multiple setup locations may be used to gather the necessary data for generating the 3D point cloud. Clicking on a Setup will load the 3D point cloud
GeoTags	This tab provides a list and downloadable versions of the GeoTags that are located within the setup. Also provided is the name of the GeoTag, type of tag, and images showing where the GeoTag is visible. The downloadable version is in CSV format with the following information: Name, Category, Label, Link, Type, PosX, PosY, PosZ, visiByids, visiBynames. The PosX, PosY, PosZ data relate to the 3D point cloud coordinates of the GeoTag. visiByids and visiBynames relate to the scan ID and the name of the scan as it is saved to the server.



---

Snapshots	As indicated by the name, this tab includes 2D digital images captured within the TruView environment, similar to a screenshot of the scene that is being viewed. This could be useful for an operator to reference a certain asset or area of the plant that needs further inspection or attention. Markups and measurements may be made within a snapshot for future reference
Assets	This tab contains a list of assets that have been identified in the setup
Info	This tab consists of general information about the scan such as identifier (ID), name, setup Meta Data, Coordinate Systems, Panorama Meta Data, View Defaults, Resolution, and Neighbour scans

---

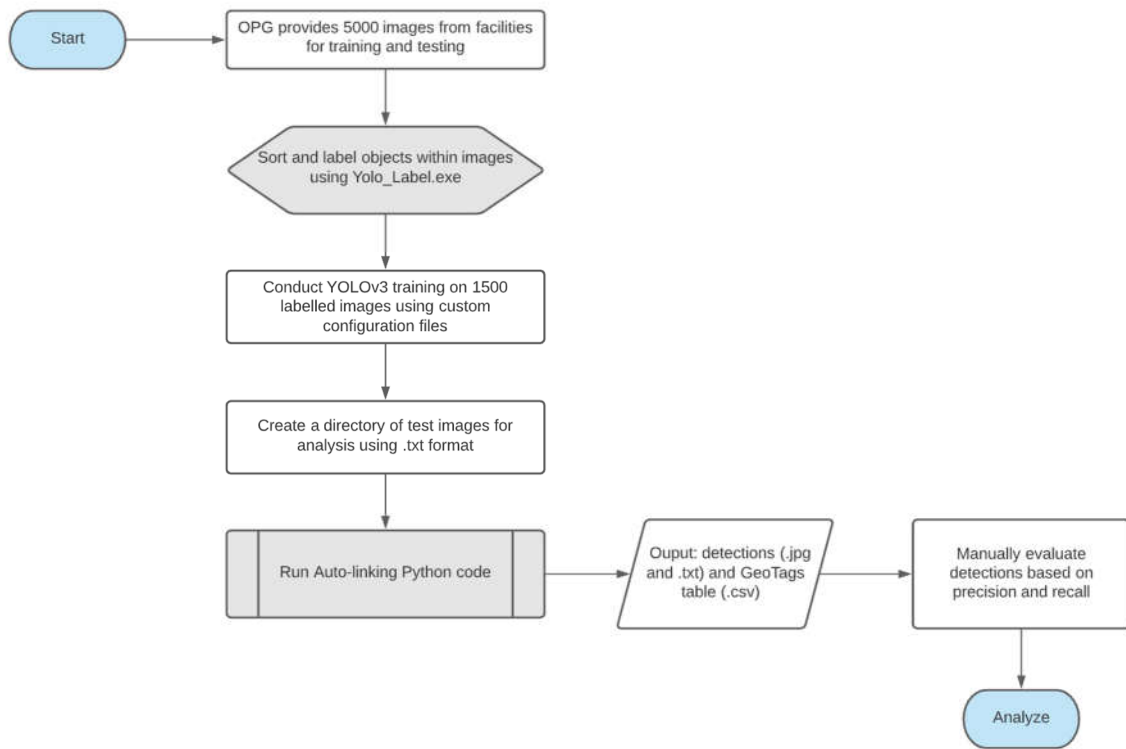
When an operator chooses to add a GeoTag in the 3D point cloud, they will select the appropriate setup, search for SSCs, click on or near the SSC and select the “GeoTags” icon from the TruView menu. This will lead to a pop-up window that requests the following information is inputted: Name (often just the equipment ID), Category (descriptive to what is being tagged, i.e., Equipment ID tag), Label, and any pertinent attachments. Once this pop-up is complete the GeoTags tab is updated, and the information is stored. Attachments that are included will be saved to the server in the scan folder that is named after the visiByids ID. The GeoTag is created based on world point coordinates, meaning it is relative to the SSC’s location in space. When SSCs are replaced or moved, it is necessary to update the location of the GeoTag itself to remain relative to the corresponding SSC. If the scan is updated without changes to the SSC and the layout of the space, the location of GeoTags does not have to updated.

It is evident from this description of the tagging procedure that there are multiple manual steps involved in this process. The most time-consuming step is searching for the SSC and ensuring that all the assets have been tagged within the 3D point cloud. This is highly time-consuming due to the high quantity of assets located within an NPP facility. The use of Auto-linking is also useful for adding

GeoTags when updating scan setups. As mentioned, if the layout of the area being scanned is not altered, the locations of GeoTags do not have to be updated. This means, with new scan setups, the same GeoTag location and information can be simply imported to reduce the manual effort of inputting GeoTags in the updated scan. The following sections detail the scope of Auto-linking and how it will improve the current procedure to be more time and cost efficient.

### **6.1.2 Scope of Auto-linking Experimental Analysis**

This experimental analysis is intended to demonstrate a real-life application of Auto-linking as it relates to the OPG procedure of tagging SSCs for asset management. To demonstrate this functionality, the Auto-linking pipeline that was explained in Chapter 5 was completed using OPG data and OPG compatible software. The data used for training the YOLOv3 object detection was retrieved from images throughout the OPG Darlington Facility. This training dataset consisted of 1500 labelled images. The object detection and OCR portion of the Auto-linking pipeline was evaluated based on 130 test images. To evaluate the entire Auto-linking algorithm, a demonstration was completed on four scan setups provided by OPG. The reason for evaluating the pipeline in this manner was due to the limited availability of scan data that could be shared under security restrictions. The workflow of the Auto-linking analysis is shown in Figure 6.1 and will be described in detail in later sections.



**Figure 6.1 Workflow of conducting the Auto-linking analysis to evaluation object detection and OCR output**

The results of the object detection and OCR are evaluated according to level of accuracy in detection and OCR. A confusion matrix is used to quantify the results of the object detection. The evaluation of running the four scan setups through the Auto-linking algorithm is intended to demonstrate the full application. The evaluation will also provide constructive feedback on how the pipeline could be improved and how the automatic detection should be validated.

### 6.1.3 Assumptions and Limitations

The primary assumptions and limitations that relate the results of this analysis correspond with the choice of object detection algorithm and training method. YOLOv3 was chosen for the Auto-linking object detection because of its efficiency and ease of implementation. However, there are other deep learning-based algorithms available that are also suitable for this type of detection. The results of this analysis are also limited by the training dataset. Factors that affect the results of the object detection training revolve around the quantity and quality of the training images. As previously mentioned, the preliminary training was conducted on a dataset of 500 images. Based on the results of this training it

was determined that a larger dataset of approximately 1500 images would provide higher detection accuracy and improved results. The size of this dataset was limited to account for computer processing capacity. The dataset included images from multiple locations throughout the OPG facility to account for variation in quality of images in attempt to improve accuracy. Training a dataset with images that have some variability in environment such as lighting, resolution, backgrounds, can benefit the detection performance.

The demonstration of the full Auto-linking algorithm was limited by the availability of the scan data that could be provided. Since the code that is used to overlay cube maps over point clouds is not available to the public, the raw data had to be converted by Leica internally to provide the 16-bit images, offset, and scale, that would be used to convert 2D pixel coordinates to point coordinates. However, due to security law that prevents the sharing of facility data to the United States, the data was restricted to scan setups within the training facility at OPG. This reduced the amount of scans that could be used for the evaluation. However, since the Auto-linking algorithm is largely dependent on the object detection and OCR, it was determined that the analysis would primarily focus on the evaluation on a set of image data.

#### **6.1.4 Description of Process, Tools, and Software**

The experimental analysis was conducted on a computer with a Windows operating system. The methodology followed for this analysis was described in Chapter 5, and the workflow is shown in Figure 6.1. First, the training images were labelled using Yolo\_label.exe with three classes; Equipment Tag A, Equipment Tag B, and Equipment Tag C to categorize the three main tag formats observed in the facility. Figure 6.2 provides an example of each class.



**Figure 6.2 Examples of the three classes. Left to right: Equipment Tag A, Equipment Tag B, and Equipment Tag C**

The labelling of the training images was done manually by identifying visible Equipment ID tags within images of the OPG facility. The Equipment ID tags that were labelled for training involved those that were visible by the human eye with relatively high resolution and visibility. The bounding boxes were created closely along the bounds of the Equipment ID tag to reduce interference of the background environment. It was important to label the Equipment ID tags with a high level of precision because the quality of labelling impacts the overall detection accuracy. Once the labelling of the test dataset was complete, it was necessary to prepare for the object detection training. The first requirement is the installation of Darknet (Redmon, 2016), which is the base framework for running the YOLOv3 algorithm. The repository used was a Windows compatible version of Darknet for YOLOv3 by Alexey AB (AB, 2019). The software requirements for running this repository are listed below:

- Windows or Linux
- CMake – version 3.12 or greater
- Visual Studio
- CUDA – version 10.0 or greater
- OpenCV – version 2.4 or greater

- cuDNN – version 7.0 or greater
- GPU with CC of 3.0 or greater
- GCC or Clang for Linux and MSVC 2017/2019 for Windows

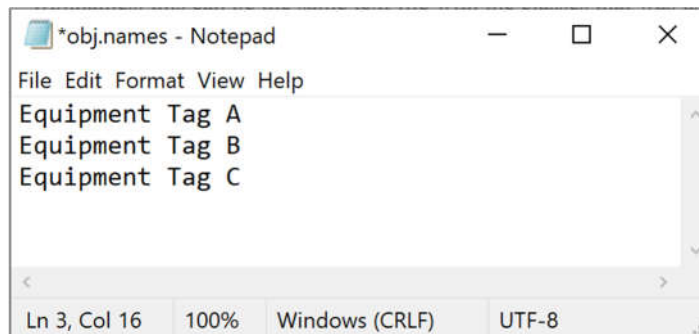
Running the algorithm with a GPU and/or the CUDA accelerates the training processes and is recommended for reducing the overall time and accuracy of object detection. OpenCV allows for viewing object detections in real time rather than saving the file and opening separately. Once these requirements have been met, it is necessary to build Darknet from source. This makes it possible to execute the YOLOv3 commands for object detection. Prior to building darknet, a pre-trained convolutional weights file was downloaded from PJ Reddie (Redmon, 2016) and saved to the same folder that contains the darknet repository. The next step involved setting the parameters in the Makefile that would determine how darknet is installed. This Makefile can be manipulated to choose whether to run the darknet with GPU, CUDA, OpenCV and cuDNN. For this analysis GPU, CUDA, OpenCV and cuDNN were initiated. After running the Makefile in the Windows Powershell, darknet was successfully configured.

The first step to training is creating a folder with the training dataset. This folder was called “OPGobj\_data” and contained the images and text files from the labelling process. Once the data folder was compiled, it was necessary to customize the configuration file for YOLOv3. The custom changes made relate to the classes specified for the custom dataset and the computer processing. The modifications were as follows:

- $n = \text{Number of classes} = 3$
- $\text{Max\_batches} = 2000 * n = 6000$ : Number of times the training will run through the dataset (with minimum of 4000)
- $\text{Steps} = 0.8 * \text{Max\_batches}, 0.9 * \text{Max\_batches}$
- In each [yolo] layer:  $\text{classes} = n = 3$
- In each [yolo] layer:  $\text{random} = 1 = 0$ 
  - If  $\text{random} = 1$ , the training will resize the image as it trains. Changing this to 0 prevents an out-of-memory error when running the training.
- In each [convolutional] layer before the [yolo] layer:  $\text{Filters} = (n + 5) * 3 = 24$

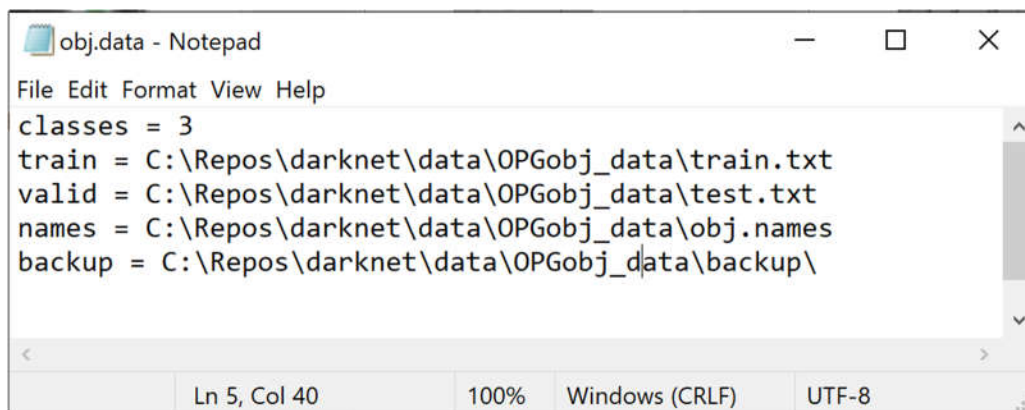
After customizing the configuration file, it was necessary to create the custom object files to specify the location of training data, testing data, and parameters. In the same folder that the data is stored, the following files were created:

- **Obj.names** (Figure 6.3): this can be the same text file with the classes that was used to label the data. It will simply consist of a list of the class names, one per line (Equipment Tag A, Equipment Tag B, Equipment Tag C)



**Figure 6.3** The **obj.names** file that was used to identify object labels

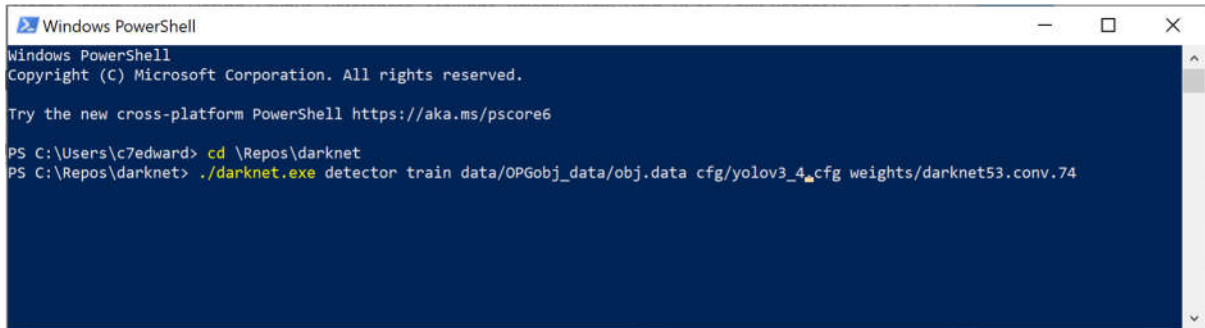
- **Obj.data** (Figure 6.4): this will specify the quantity of classes, and location of training data, test data, **Obj.names**, and a backup folder for saving the custom weights.



**Figure 6.4** The **obj.data** file for training and testing

To specify the training data and test data, a text file is required for both. This can be created using a Python code that runs through the saved images and lists the file names in a .txt document. These are then the files referenced in the **Obj.data** file. The test data consisted of approximately 150 images from the image dataset provided by OPG.

Once these steps have been accomplished, the final step was running YOLOv3 with the pre-trained convolutional weights that were previously downloaded. The training command prompt was used to run the custom YOLOv3 training in Windows Powershell as shown in Figure 6.5.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\c7edward> cd \Repos\darknet
PS C:\Repos\darknet> ./darknet.exe detector train data/OPGobj_data/obj.data cfg/yolov3_4.cfg weights/darknet53.conv.74
```

**Figure 6.5 Command prompt that runs the YOLOv3 training for custom dataset using pre-trained weights**

Once the detector successfully finished training, a Python script was developed to run the detector on the images in the test directory. This script was included in the general Python code as described in Chapter 5 and is shown in Appendix B.

The final step of the Auto-linking analysis was evaluating the testing dataset using the complete Python code which runs the detector, runs OCR, computes the 3D point coordinates, and exports the GeoTag information. The exported information was then converted to a suitable format to import directly into the 3D point cloud software, Leica Cyclone. The following section will describe the metrics used to evaluate this analysis.

### 6.1.5 Evaluation Metrics

As stated above, approximately 150 test images were run through the Auto-linking algorithm to evaluate the object detection and OCR accuracy. In addition to this analysis, four scan setups were provided by OPG to demonstrate the complete process using Leica software and related tools.

The metrics used to evaluate Auto-linking include one metric for the object detection and another metric for the OCR and information sorting, as stated below:

1. Correctness of OPG Equipment ID Tag detection: this metric relates to the number of True Positives (TP), False Positives (FP) and False Negatives (FN) in the test dataset. This metric relates to the Equipment tags per image. A TP is defined as an accurate detection of an Equipment ID tag, including the class (Equipment Tag A, Equipment Tag B, or Equipment



Tag C). An FP relates to areas of the image that were detected as an Equipment ID tag but are not. An FN is when an actual Equipment ID tag is not detected. Lastly, True Negatives (TN) were not quantifiable, since it would apply to the entirety of the image without tags. These values were displayed in a confusion matrix along with the precision and recall values.

2. Accuracy of the OCR and sorted GeoTag information: this metric relates to the information that is exported and then imported to label GeoTags. The accuracy of the OCR is important for precise labelling of assets for facility management. The accuracy of the OCR is measured by searching a database of the existing Equipment ID tags in the test images and comparing to the OCR results. The sorted GeoTag information is evaluated by ensuring the necessary data is sorted into the corresponding columns.

The selection of these metrics is based on discussions with OPG and current procedures that OPG implements. Firstly, it is critical to have the correct Equipment ID labelled within the 3D point cloud to associate the correct data and management documents with the SSC. Methods of validating the equipment ID were presented in Chapter 5 and will be incorporated into the analysis for the purpose of evaluation. Secondly, it is necessary to have the GeoTag in a reasonable proximity to the SSC within the 3D point cloud. This is necessary because operators will likely search for the SSC itself rather than the corresponding GeoTag. Therefore, evaluation of the bounding box coordinates of the actual Equipment ID tag is advantageous to ensure the GeoTag would be located within an adequate proximity to the SSC. Once the SSC is visually located, the GeoTag can be clicked to access any information that the operator requires. This metric is not as critical in comparison to the first as it is not required to have the GeoTag located precisely on the SSC.

### **6.1.6 Results of Auto-linking Analysis**

The output of running the Python script that enables the YOLOv3 detector, OCR, and GeoTag information includes the following:

1. Images (.jpg) that show the detected bounding boxes labelled with the object classes. A subset of the detection images is shown in Appendix B for reference.
2. Text files (.txt) that include the detection time in milliseconds, the detected object classes, the accuracy in percentage, and bounding box coordinates

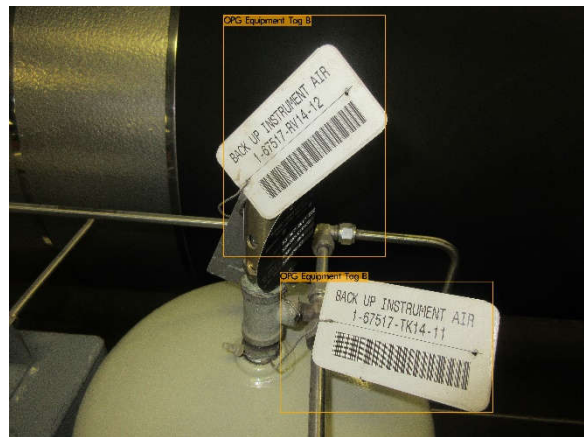
- An output.csv file that includes the GeoTag information in a table format with the appropriate headers for Geo-Tagging ("image", "name", "category", "label", "link", "type", "posx", "posy", "posz", "visiByids", "visiBynames", "originalids")

In addition to these output files, an important file is the database that includes the existing Equipment ID tags in the test dataset. This was created manually for the purpose of comparing the detected objects with the actual conditions. The database consisted of a list of the Equipment IDs for each image that was run through the Auto-linking. This database is included in Appendix C.

This section will begin by previewing results from the Auto-linking analysis, including the outputs listed above. Figures 6.6 (a) through (d) are example images of the detections for each class type. These images show how the Equipment ID tags were detected by bounding boxes with the corresponding class label.



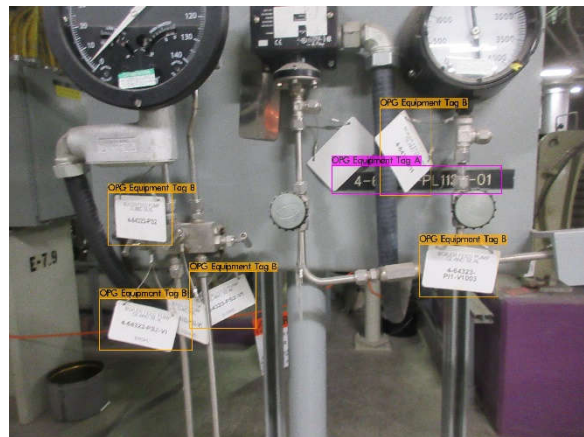
(a)



(b)



(c)



(d)

**Figure 6.6 Examples of detections for each class. (a) OPG Equipment Tag A detection, (b) OPG Equipment Tag B detections, (c) OPG Equipment Tag C detection and (d) Combination of OPG Equipment Tag A and OPG Equipment Tag B detections**

The corresponding text files for each of the above examples are shown in Figures 6.7 (a) through (d). As shown, the time taken for detection is provided, and the object class label is listed along with detection accuracy and bounding box coordinates. The detection accuracy that is listed in this output text file is computed by YOLOv3 algorithm and is not explicitly used to evaluate the results of the overall detection accuracy since it is specific to each detection. These coordinates are later used to transform the pixel location of the detection Equipment ID tag into 3D coordinates to import into the point cloud.

```

IMG_1278.txt - Notepad
File Edit Format View Help
net.optimized_memory = 0
mini_batch = 1, batch = 16, time_steps = 1, train = 0

seen 64, trained: 96 K-images (1 Kilo-batches_64)
data/OPGobj_data/IMG_1278.JPG: Predicted in 1194.713000 milli-seconds.
OPG Equipment Tag A: 95%      (left_x: 2213  top_y: 1620  width: 742  height: 128)

```

(a)

```

IMG_1246.txt - Notepad
File Edit Format View Help
net.optimized_memory = 0
mini_batch = 1, batch = 16, time_steps = 1, train = 0

seen 64, trained: 96 K-images (1 Kilo-batches_64)
data/OPGobj_data/IMG_1246.JPG: Predicted in 1070.990000 milli-seconds.
OPG Equipment Tag B: 91%      (left_x: 1926  top_y: 83  width: 1459  height: 2174)
OPG Equipment Tag B: 95%      (left_x: 2433  top_y: 2472  width: 1921  height: 1181)

```

(b)

```

IMG_1240.txt - Notepad
File Edit Format View Help
net.optimized_memory = 0
mini_batch = 1, batch = 16, time_steps = 1, train = 0

seen 64, trained: 96 K-images (1 Kilo-batches_64)
data/OPGobj_data/IMG_1240.JPG: Predicted in 1295.979000 milli-seconds.
OPG Equipment Tag B: 86%      (left_x: 284  top_y: 2271  width: 143  height: 120)
OPG Equipment Tag B: 89%      (left_x: 988  top_y: 1911  width: 157  height: 108)
OPG Equipment Tag B: 57%      (left_x: 1044  top_y: 2431  width: 66  height: 112)
OPG Equipment Tag C: 99%      (left_x: 2239  top_y: 1939  width: 1537  height: 365)

```

(c)

```

IMG_1277.txt - Notepad
File Edit Format View Help
net.optimized_memory = 0
mini_batch = 1, batch = 16, time_steps = 1, train = 0

seen 64, trained: 96 K-images (1 Kilo-batches_64)
data/OPGobj_data/IMG_1277.JPG: Predicted in 1045.367000 milli-seconds.
OPG Equipment Tag B: 93%      (left_x: 813  top_y: 2623  width: 833  height: 473)
OPG Equipment Tag B: 87%      (left_x: 887  top_y: 1676  width: 580  height: 478)
OPG Equipment Tag B: 69%      (left_x: 1622  top_y: 2383  width: 597  height: 515)
OPG Equipment Tag A: 82%      (left_x: 2896  top_y: 1429  width: 1531  height: 259)
OPG Equipment Tag B: 67%      (left_x: 3332  top_y: 913  width: 465  height: 790)
OPG Equipment Tag B: 100%     (left_x: 3684  top_y: 2136  width: 709  height: 475)

```

(d)

Figure 6.7 Text files that correspond with the detections shown in Figure 6.6

The output.csv file containing the full list of information related to each detected equipment ID tag is included in Appendix D. This output file could be manipulated by a human operator to adjust the OCR read equipment ID tags, or other information that is included. Since many of the columns are related to scan ID unique identifiers and other scan related information, a standard name was given for the purpose of the Auto-linking analysis. The takeaway from this output file is the ability to sort the detected Equipment ID tags and related information into a format that can be imported into the asset management tool with minimal adjustments and effort.

After obtaining these output files, it was necessary to compare the detections to the actual conditions. This was completed by evaluating each image to determine the TP, FP, and FN values for each test image. Of the 150 test images there were 313 tags present and visible to the human eye. In comparison, the Auto-linking detection accurately detected 212 Equipment ID tags, incorrectly detected 21 Equipment ID tags and a total of 101 Equipment ID tags went undetected. These values, as well as the precision and recall values are shown in Table 6.2. The equations for the precision and recall values are included in the confusion matrix.

**Table 6.2 Confusion matrix for the Auto-linking object detection**

<b>Confusion Matrix</b>	Detected Tag	Undetected Tag	<b>Recall = <math>\frac{TP}{TP+FN}</math></b>
Actual Equipment ID Tag	212 [TP]	101 [FN]	0.68
Not Equipment ID Tag	21 [FP]	N/A [TN]	
<b>Precision = <math>\frac{TP}{TP+FP}</math></b>	0.91		

The precision value for this testing was relatively high with a value of 0.91, indicating that when a detection was made it was most probably an Equipment ID tag. The recall value represents the sensitivity of the detection by quantifying the portion of actual Equipment ID tags that were detected. This indicates that approximately 30% of the Equipment ID tags went undetected during the analysis. For an object detection algorithm, these results are favourable and are a considerable improvement from the previous methods attempted for this application.

To understand why the 101 Equipment ID tags were not detected, further data analysis was conducted to document the condition of the image as well as the condition of the tag, and corresponding environment. Conducting this analysis provides perspective as to why certain tags were not detected by the algorithm and could lead to insight into how to improve the detector. The following parameters were considered as possible reasons an Equipment ID tag went undetected:

- **Resolution:** poor resolution caused by an overall blurry image or due to far distance from the camera viewpoint could distort the Equipment ID tag and reduce the detection ability. This also reduces the accuracy of the OCR because it becomes more challenging for the text detector to read text in a poor resolution image. Figure 6.8 (a) and (b) show two instances where the resolution was very poor causing blurring of the Equipment ID tag text and some surrounding.



**Figure 6.8 Poor resolution as demonstrated by these images could impact the detector accuracy  
(a) IMG\_1211.JPG and (b) IMG\_1329.JPG**

- **Covered:** this parameter applies to Equipment ID tags that are covered by equipment or other tags, Equipment ID tags that are blank because they are flipped over, Equipment ID tags that are bent or Equipment ID tags that are cut off due to the bounds of the image. This should be considered because the training was conducted on primarily full and visible Equipment ID tags therefore there are fewer instances that allow the detector to learn that a partial Equipment ID tag still qualifies as a detection. Figure 6.9 (a) shows an image with three Equipment ID tags, however one is not detected since it is covered by equipment and another is not detected since it was partially cut off by image boundary. The third tag was still detected despite being cut off which shows the detector will still detect some



Equipment ID tags that exhibit this parameter. Figure 6.9 (b) shows two Equipment ID tags and one was not detected likely due to the fact that it is flipped over and blank.



**Figure 6.9 Instances where coverage causes an Equipment ID tag to be undetected. (a) IMG\_2340.JPG contains one tag covered by equipment and another that was cut off by image boundary (b) IMG\_1206.JPG shows a blank tag that was not detected**

- **Lighting:** this is an important parameter to consider for object detection because extreme lighting conditions (very dark or very bright) can affect the boundaries of an object that are often used as identifying properties. Also included by this parameter are Equipment ID tags that have backgrounds with similar colour or texture or backgrounds of extreme contrast to the typical environment. Figure 6.10 (a) shows an image with extremely poor lighting which caused the Equipment ID tag that is barely visible to go undetected. Figure 6.10 (b) shows an instance where the Equipment ID tags have similar colour to the background components due to brighter lighting which likely caused them to go undetected.



**Figure 6.10 Undetected Equipment ID tags due to lighting (a) IMG\_1288.JPG and (b) IMG\_1247.JPG**

- Damage:** this parameter considers the physical condition of the Equipment ID tag that could interfere with detection because the damage has caused the Equipment ID tag to look different from a normal state. Damage may include ripping of the tag, sever discoloration, or missing text. Figure 6.11 (a) shows an Equipment ID tag that was damaged by scraping and bending which altered the appearance and likely caused the false negative detection. Figure 6.11 (b) shows an instance where an Equipment ID tag is discolored and thus altered the appearance of the Equipment ID tag.

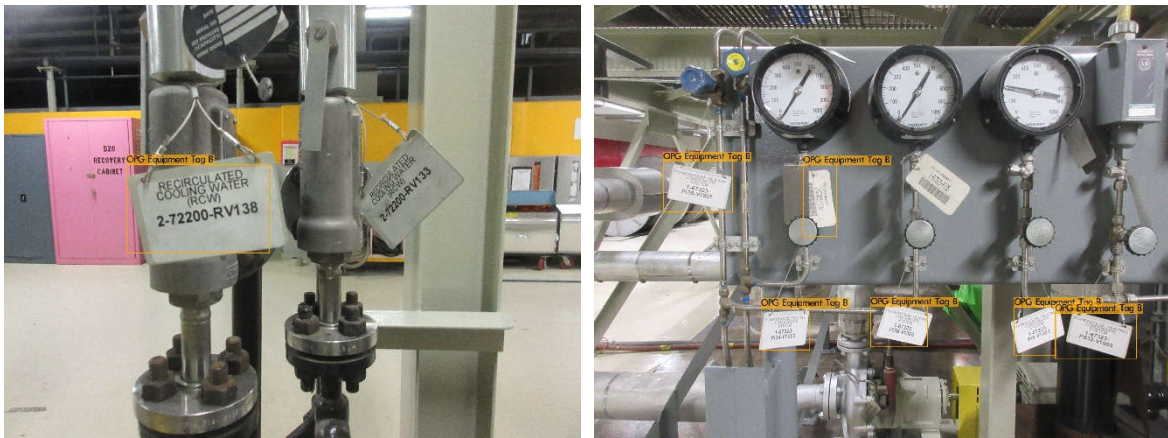


**Figure 6.11 Instances where Equipment ID tags are not detected due to damage to the object itself (a) IMG\_1167.JPG shows damage to the text and (b) IMG\_1242.JPG shows discoloration**

- Normal:** this parameter was used to identify Equipment ID tags that were not detected but did not appear to have any negative factors such as those listed above. This parameter



indicates the Equipment ID tag was undetected simply due to inaccuracy of the detector. Object detectors are not 100% accurate which explains why there are some tags that were not detected. Figure 6.12 (a) and (b) show examples of when other Equipment ID tags are detected however one goes undetected despite the lack of visual discrepancy such as damage, poor resolution, poor lighting, or coverage. These instances are not of high concern, because if a human operator verifies the neighbouring detections, it will be simply to identify the Equipment ID tags that were missed.



**Figure 6.12 Examples of when normal Equipment ID tags are undetected despite lack of visual discrepancy to the standard format (a) IMG\_1205.JPG and (b) IMG\_1241.JPG each have one false negative**

These parameters were considered when reviewing the resulting image detections and they were documented along with the count of TP, FP, and FN. The reasons are included in the second table presented in Appendix B. Table 6.3 below summarizes the count of each parameter and compares the values to the total of undetected Equipment ID tags (FN) and the total amount of actual Equipment ID tags.

**Table 6.3 Possible reasons for undetected Equipment ID Tags and Comparison to Total Values**

Possible Reason	No. FN per reason	% Undetected (101 Tags Undetected)	% Total (313 Actual Tags)
Resolution	20	20%	6%
Covered	26	26%	8%
Lighting	30	30%	10%
Damage	12	12%	4%

Normal	13	13%	4%
--------	----	-----	----

The importance of completing this data analysis is rooted in understanding what will improve the Auto-linking analysis detection. These values show that the undetected Equipment ID tags without visible discrepancies (i.e. Normal) were of the lowest undetected parameters making up of just 4% of the Equipment ID tags within the test dataset. It is evident from these values that lighting had the greatest impact for undetected objects. This is likely due to the effect extreme lighting, either very dark conditions or very bright conditions can have on changing the appearance of an object in an image. The solution to experiencing lower detection accuracy due to these possible reasons is to expand the training dataset if there is processing capacity available or improving the labelling of data that is used for training. Further discussion of these results is included in Chapter 7.

### 6.1.7 Correction of Auto-linking OCR

As previously indicated, there are various methods for validating the OCR that was outputted from the Auto-linking analysis. To automate the process of validation, a Python script was developed that uses the Jaccard distance to determine the correct Equipment ID tag based on the OCR output. The script for this correction algorithm is included in Appendix E.

This code imports both the existing database of Equipment ID tags and the output.csv file from the Auto-linking analysis. It identifies the list of the OCR detected text through calling the correct column from the output.csv file. Then, a function was defined called “getcorrections()” which determines the Jaccard distance for the OCR entries compared the database entries. The database entry with the lowest Jaccard distance is selected as the correction value. The output of this process is a list of the corrected Equipment ID tags which will improve the accuracy of the labels that are imported into the point cloud from the .csv file. An example of the original OCR text, the corrected text, and the actual Equipment ID text is shown in Table 6.4 below.

**Table 6.4 Comparison of Original OCR Text to Corrected and Actual Text**

Image	OCR Text	Corrected Text	Actual Text
IMG_1127.JPG	7111scl	7111-sc2	7111-sc2
IMG_1127.JPG	7111scl	7111-sc2	7111-sc2
IMG_1205.JPG	272200rv	2-72200-rv138	2-72200-rv138
IMG_1206.JPG	2722007134	2-72200-v134	2-72200-v134
IMG_1207.JPG	2122007132	2-72200-v132	2-72200-v132

IMG_1207.JPG	2122007132	2-72200-v132	2-72200-v132
IMG_1207.JPG	2122007132	2-72200-v132	2-72200-v132
IMG_1207.JPG	2722001137	2-72200-v137	2-72200-v137
IMG_1207.JPG	2722001137	2-72200-v137	2-72200-v137
IMG_1208.JPG	2161220tw	2-67220-tw151	2-67220-tw152
IMG_1208.JPG	222ow	2-722	2-67220-tw151
IMG_1209.JPG	172200rv1336	2-72200-rv133	1-72200-rv138
IMG_1212.JPG	172100v7011	1-72100-v3013	1-72100-v3013
IMG_1214.JPG	161517	151	1-67517-prv31-4
IMG_1215.JPG	167220p15111	1-67220-pi51	1-67220-pi51
IMG_1215.JPG	167220p151v1	1-67220-pi51-v1	1-67220-pi51-v1
IMG_1215.JPG	167220p5111	1-67220-pi51	1-67220-pi51-v1
IMG_1215.JPG	167220p1511	1-67220-pi51	1-67220-pi53
IMG_1215.JPG	167220p1511	1-67220-pi51	1-67220-pi53
IMG_1215.JPG	167220p1511	1-67220-pi51	1-67220-pi53-v1

The full output file contains the corrected entries and actual entries, as shown in the first table presented in Appendix E. The text corrector adjusts the text into the correct format, however, it does not perform as well as a human operator. It is therefore recommended that a second stage of validation of the Equipment ID tag text is completed at the end stage when the GeoTag is created.

### 6.1.8 Mapping Detections to Point Coordinates

The process for determining the point coordinates based on the image coordinates of the detected Equipment ID tag was described in Chapter 5. To implement this process, a Class was defined in Python called “CoordsMapping”. By defining this class the mapping procedure could be easily called within the Auto-linking algorithm to compute the point coordinates. The script for CoordsMapping is included in Appendix B following the Auto-linking script. These point coordinates are included in the output table such that they can be imported into the point cloud to create the GeoTag in the target location. The 2D to 3D coordinates mapping was tested on a small dataset of scan setups from the OPG training facility. An example is shown below where the same OPG Equipment ID tag was visible in two separate scan setups. This Equipment ID tag was detected, processed through OCR and CoordsMapping, and the output showed that two similar sets of world point coordinates. This indicated that the world coordinates are relatively accurate and a GeoTag would be automatically linked for this Equipment ID tag in an adequate location within the point cloud.



**Figure 6.13 Demonstration of Auto-linking from Two Scan Setups**

As shown in Figure 6.13, the same Equipment ID Tag is visible from two scan setups within the training facility and was detected through Auto-linking as Equipment Tag C. Table 6.5 below shows some information from the output file to demonstrate the similarity in world coordinates. This indicates that the Equipment ID tag will be accurately placed within the point cloud.

**Table 6.5 Relevant Information for Detected Equipment ID Tags**

Image	OCR ID	Corrected ID	posx	posy	posz
TEST_TAG_-_SETUP_1_f.jpg	34710hx4	3-4110-hx5a	0.726277	1.81047	-1.58239
TEST_TAG_-_SETUP_2_f.jpg	343ti0hxa	3-43110-hx5a	0.679779	3.085517	-1.58285

However, as stated previously, further validation can be conducted by a human operator to make adjustments to the automated GeoTag, as necessary. If it is necessary to move the GeoTag for some reason, it can be done manually through TruView.

## Chapter 7

### Discussion

The work presented throughout this thesis consisted of primarily two objectives: first, the development of a DT framework specific for NPP implementation and second, the development of the Auto-linking algorithm to improve the efficiency of asset management. The discussions below evaluate both the DT framework and the Auto-linking algorithm based on feedback from industry partners and the experimental analysis that was conducted.

#### 7.1 Evaluation of the DT Framework for NPPs

The DT framework and related support documents were developed specifically for application within NPP facilities for asset management. In creating the framework, workflows, and support documents the primary sources of inspiration and feedback were literature review and industry partnership. The DT framework was developed in a layered approach to identify the steps to building a virtual and semantic model of a legacy asset. The framework combines the primary layers which were perception, communication, and action. This simplifies the definition of DTs because it breaks the process down to obtaining a digital representation of the physical environment in the perception layers, incorporating the semantic properties through operating data and computational processes in the communication layers, and lastly using the digital twin for multiple management purposes across the company network.

The seven-layer framework was tailored to NPP application through an understanding of current practices. For instance, the digital acquisition includes both imagery and point clouds, the data retrieval involves characteristics such as sensor and indicator data that can gather information about systems and sub-systems, and computation is focused on the digital database of legacy assets. Following this framework, support documents were necessary to guide the completion of the work to achieve the layers. Firstly, definitions of DT levels were developed based on current procedures and standards in NPP facilities. The levels were simplified to optimize level of effort while maintaining sufficient level of detail for operations and maintenance. An understanding of procedures and standards was gained from discussion with OPG industry partnerships about how engineers and operators would utilize a DT and what functions they should include. For example, while the traditional definition of a DT includes a geometric model, the level 1 DT defined in this thesis is simply a point cloud of the physical environment with links to the appropriate asset management tool.

This reduces the DT process for lower-level DTs, because in terms of NPP asset management, visual representation through point clouds are sufficient for the asset management objectives. This approach allows for accurate representation of the current state of the asset and can be used for measurements or mark-ups. The geometric modelling is incorporated in higher levels for specific assets that would require more detailed modelling. It was also determined through feedback and discussions that an appropriate standard throughout the facility is a level 3 DT. This is an expected level for all DTs that will be constructed, because it is a semantic model which includes geometric modelling, sensor data feedback, and links to the asset management tool without the requirement of additional computational investment for predictive maintenance or autonomous decision making. Higher levels of DTs such as level 4 or level 5 were deemed suitable for other teams within the organization that focus primarily on preventative or predictive maintenance of highly critical assets. The feedback provided by industry partners, and knowledge of NPP management through literature review was imperative to develop an effective framework.

The support documents aim to guide those implementing DTs to select the appropriate level and ensure there is consideration for factors such as criticality and safety as well as identifying limitations at early stages. Following the support documents and workflows is intended to streamline the implementation of DTs and maintain a consistent approach throughout an entire facility. Lastly, the network and connections of DTs for systems and sub-systems is dependent on the facility itself and existing facility organization. However, this is a critical step for understanding how dependent DTs interact with one another. This consideration varies from a traditional DT or BIM because commonly a DT or BIM of a full facility would implement the same level of detail throughout. However, it became apparent through this research that due to the magnitude of assets and the varying functions, it would be a better investment to select a DT level based on the SSC of interest to avoid over-modelling certain systems and sub-systems where a high functioning DT is not essential.

### **7.1.1 Limitations of DT Framework**

The primary limitation of this research is the slow adoption of DTs in NPPs which resulted in few application scenarios and use cases to guide the development of the framework. Due to this limitation, feedback was primarily provided through the industry partnership, which clearly identified specific challenges faced using current FM practices in NPP facilities. The evaluation of the DT framework was also limited by not having quantifiable measures of its applicability and functionality. To determine the applicability of the presented framework and support tools, presentations that included

detailed walkthroughs and example scenarios were held with industry partners. The constructive feedback from these presentations were incorporated throughout the development process.

## **7.2 Evaluation of Auto-Linking**

The intent of the Auto-linking algorithm was to reduce the manual effort and time required to search for and label assets within the point cloud software so that they could be linked with the asset database. The point cloud (and associated information in the TruView environment) is a fundamental component of the DT framework, because it acts as the base layer for virtual representation, can be used for detailed modelling, and contains information related to the asset's operation and maintenance. Machine-learning-based and deep-learning-based approaches were examined to automate the detection of Equipment ID tags from images and read the Equipment ID that identifies the SSC. The main disadvantage of the machine-learning-based approaches was the need for image dependent filters and parameters. This would require additional pre-processing for each batch of images which made it unsuitable for application in a facility such as an NPP due to the high number of assets and variability in lighting and resolution. The deep learning-based approach proved to be more suitable since it was trained on a high volume of images with variations of environment thus reducing false negatives and false positives. After determining that a deep-learning-based approach was most appropriate for this application of object detection, the Auto-linking pipeline was completed using Python code. The code used the detector to identify Equipment ID tags, then cropped the image based on bounding box coordinates to run through the OCR. The OCR obtained the unique identifier which was then stored in a table format along with other GeoTag information such as coordinates and scan file information. This was outputted from the code in a format that could be imported into point cloud software to create the necessary asset tags.

To quantify the accuracy of the Auto-linking algorithm, an experimental analysis was set up using data provided by OPG. The object detection and OCR were primary focuses of this analysis which included running the Auto-linking algorithm on approximately 150 test images. The evaluation of the detections showed a precision of 91% and a recall of 68%. These values are relatively high for object detection on a custom dataset with a training dataset of just 1500 images. There were a total of 313 Equipment ID tags identified in the test dataset and 212 of these tags were detected while 101 were false negatives. To further understand this amount of undetected Equipment ID tags, data analysis was conducted to evaluate possible reasons of false negatives, which included: resolution, tag coverage, lighting, damage, and normal condition. Completing the evaluation of possible reasons

identified that the primary cause of the undetected tags were visual changes to the Equipment ID tag that reduced its similarity to the standard format seen in training. For example, the poor resolution blurred the edges of the tag making features less recognizable for the detector, high contrast or extremely low contrast caused by variability in lighting could also impact the detection, and lastly damage could change the colour or shape of the Equipment ID tag. It was also important to determine the amount of false negative detections that were related to Equipment ID tags with no visual discrepancy since that indicated inaccuracy of the detector itself. It was found that these normal tags that were undetected made up of 4% of the total amount of the test dataset. This value is very low and shows that the detector performed successfully.

### **7.2.1 Combination of Auto-Linking and Human Operator**

Given the results of the Auto-linking analysis, it is evident that there was high performance of the object detector for this application. However, the analysis also showed that the OCR results exhibited lower accuracy. The most dependable solution to this inaccuracy is including a human operator for verification of Auto-linking. The human operator would monitor that the correct Equipment ID tags are being imported by using a search and match tool that compares the OCR with an existing database. The human operator can also add GeoTags for the few assets that were not detected using the algorithm. The combination of Auto-linking and Human Operator would optimize the process by improving the efficiency of Geo-Tagging while maintaining a high level of verification through human input.

### **7.2.2 Limitations of Auto-Linking Algorithm**

The Auto-linking algorithm that was developed as part of this thesis is highly dependent on the deep learning object detection of equipment ID tags in images. The deep learning training was done on a dataset of approximately 1500 images that included the current equipment ID tag formats that are used within the OPG NPP facilities. Therefore, if there are changes to the standard format of equipment ID tags, it will be necessary to conduct additional training to include the changes. This training may be done with the current weights that were used in this training. The weights file will update based on the additional training images to include additional formatting. This is considered a limitation as it is unknown how the equipment ID tag format will change and will require someone with experience in object detection training to conduct this process.



The accuracy of the object detection was also limited by the training dataset that was used. Approximately 1500 images with diverse conditions were used for the YOLOv3 training to ensure various environments were considered to improve the accuracy of the object detection. It was also important to include images where the object was partially covered, blurred, or in different orientations. Despite the attempt to consider these variations in training it was evident from the analysis that certain equipment ID tags were not detected due to variability in appearance from these factors. Further training of more than 1500 images may be conducted to increase the accuracy of detection, if required. However, as mentioned in the analysis, the addition of a human operator to monitor undetected equipment ID tags would improve the overall detection while still increasing the efficiency of the procedure itself.

## Chapter 8

### Conclusion and Future Work

The recent advancements in digital data and technology have made the use of Digital Twins (DTs), Building Information Models (BIM), and Cyber Physical Systems (CPS) a predominant tool for infrastructure and facility management. While these models are commonly used to compare as-designed state to as-built state, there is a clear need for support to implement the digital technology in operation and maintenance phases for legacy assets. Specifically, facilities such as Nuclear Power Plant (NPP) facilities would benefit from DTs due to their need of up-to-date asset data for continuous real-time monitoring. This thesis explored the need for DT support through the development of a DT framework and Auto-linking algorithm for efficient legacy asset management. The intent of the DT framework is to identify the connections between the physical asset, digital data acquisition, technology, and the final digital twin. The related support documents were developed with the insight from industry partners to incorporate the process into current practices for streamlined integration of DTs. Auto-linking was developed to increase the efficiency of digital database logging and labelling assets within 3D point clouds, which is currently a manual and time-consuming process. The goal of the Auto-linking algorithm is to introduce automatic processes into the current facility management procedures to improve overall efficiency and productivity.

As presented, the levels of DT were redefined by simplifying the basic description of a DT. This redefinition was intended to coordinate levels of DTs with existing standards that are practiced in NPP facility management (FM). The simplification of the DT levels reduces the modelling investment since the basic level solely requires the point cloud and linked asset management tool. Higher levels of DT include more functions that lead to a semantically rich DT. The highest level of DT is intended for predictive maintenance and autonomous decision-making that could be used for highly critical assets with substantial design and operating data. The appropriate level of DT is mainly chosen based on a set of criteria including availability of documentation, criticality rating of the asset, and preliminary investigation. These criteria identify information that will be used for geometric modelling, the operational data, safety or critical concerns, and any restrictions that might be faced. A certain level of DT may also be chosen due to the objective of the DT. For example, if it is required for preventative maintenance, a higher level of DT would be selected despite the lack of support documentation or low critical rating. In these scenarios, it may be required to obtain additional data to

complete the high functioning DT. The support documents for the DT framework outlines these decision-making processes to advise engineers and operators in the DT construction process.

The Auto-linking algorithm was developed as a component of the DT framework to automate the process of labelling and storing digitized assets in point clouds. The existing procedure requires manual effort to detect the assets, insert a tag, identify the unique identifier and type in related information. Through automating the process, the Auto-linking algorithm was considered an essential link in the communication layer of the overall framework. The algorithm incorporated deep learning-based object detection and Optical Character Recognition (OCR) to first detect equipment ID tags that indicate an asset, then read the information pertaining to the asset. A Python code compiled the steps of the pipeline to organize and store the data in a format that can be imported into the point cloud software. Once imported, a digital tag is created within the point cloud to automatically identify assets according to the equipment ID tag. The Auto-linking algorithm successfully demonstrated increased efficiency in identifying and labelling the digitized assets. The validation of the Auto-linking algorithm can be completed by comparing the OCR results with existing database of the assets. Validation will also be completed via human operator to ensure all necessary legacy assets have been tagged within the point cloud.

## **8.1 Future Work**

The work presented in this thesis identified how the current FM practices in NPP facilities could be improved using DT and automated processes. While the support tools and recommendations that were made in this thesis show a considerable improvement in efficiency, there is room for improvement and additional research. Future work to further develop the use of digital technology in NPP FM should explore further automation of manual processes and improvement of the object detection in both images and point clouds. It would also be beneficial to examine use cases of the DT framework under various assets and applications to identify challenges that should be mitigated for more fluency of employment. The use cases for the DT framework would identify minor challenges or obstacles that should be improved for full efficiency.

After implementing the framework for various assets, and addressing these challenges, the framework and support documents could be developed into a standard document to maintain consistency across NPP facilities. It would also further define requirements for deciding on levels of DT, what support documents are needed, and how to conduct a preliminary investigation. This future

work is dependent on the results on implementation, and the integration of DT for NPP facilities will become more efficient with repeated integration.

Further research in the detection of objects within point clouds would advance the Auto-linking algorithm by implementing the object detection directly from the scan data. This would reduce the amount of pre-processing by using the raw scan data (point clouds) rather than obtaining the images for the object detection. This research could also lead to the use of object detection with real time scan data to further the efficiency of creating labelled assets within a point cloud. Additionally, object detection that identifies specific legacy assets would lead to precise detections. This would require a higher level of computing and investment, however for NPP facilities this application could improve the overall DT framework. The OCR that reads equipment ID tags could be improved by training a text detector on the format that is used throughout the facility. Lastly, to improve the Auto-linking as it is presented, additional training of the object detector could be conducted. This could be done with other object detectors or simply additional training images for improved accuracy.

## References

- AB, A. (2019). Windows and linux version of darknet. Retrieved from <https://github.com/AlexeyAB/darknet>
- Ardito, L., Petruzzelli, A. M., Panniello, U., & Garavelli, A. C. (2018). Towards industry 4.0.
- Arup. (2019). *Digital twin towards a meaningful framework*. (). London: Arup.
- Ayani, M., Ganebäck, M., & Ng, A. H. C. (2018). Digital twin: Applying emulation for machine reconditioning. *Procedia CIRP*, 72, 243-248. doi:<https://doi.org/10.1016/j.procir.2018.03.139>
- Bellekens, B., Spruyt, V., Berkvens, R., Penne, R., & Weyn, M. (2015). A benchmark survey of rigid 3D point cloud registration algorithms. *Int.J.Adv.Intell.Syst*, 8, 118-127.
- Borrmann, A., König, M., Koch, C., & Beetz, J. (2018). Building information modeling: Why? what? how? *Building information modeling* (pp. 1-24) Springer.
- Braun, A., & Borrmann, A. (2019). *Combining inverse photogrammetry and BIM for automated labeling of construction site images for machine learning* Elsevier.  
doi:10.1016/j.autcon.2019.102879
- Brilakis, I., & Haas, C. T. M. (2019). *Infrastructure computer vision* Butterworth-Heinemann.
- CANDU Owners Group. (2019, June 20,). International community converges on the future of nuclear. *COG News*, Retrieved from <http://www.candu.org/COGNews/International%20community%20converges%20on%20the%20future%20of%20nuclear.pdf#search=digital%20twins>

- Chou, T., Ho, C., & Kuo, Y. (2015). QR code detection using convolutional neural networks. Paper presented at the *2015 International Conference on Advanced Robotics and Intelligent Systems (ARIS)*, 1-5.
- Corneli, A., Naticchia, B., Carbonari, A., & Bosché, F. (2019). Augmented reality and deep learning towards the management of secondary building assets. Paper presented at the *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, , 36 332-339.
- Darrington, J. W., Browning, J. M., & Ritter, C. S. (2020). Deep lynx: Digital engineering integration hub. *Deep Lynx: Digital Engineering Integration Hub*, Retrieved from <https://github.com/idaholab/Deep-Lynx>
- Delgado, J. M. D., Oyedele, L., Ajayi, A., Akanbi, L., Akinade, O., Bilal, M., & Owolabi, H. (2019). Robotics and automated systems in construction: Understanding industry-specific challenges for adoption. *Journal of Building Engineering*, 26, 100868.
- Grieves, M., & Vickers, J. (2017). Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems. *Transdisciplinary perspectives on complex systems* (pp. 85-113) Springer.
- Hansen, D. K., Nasrollahi, K., Rasmussen, C. B., & Moeslund, T. B. (2017). Real-time barcode detection and classification using deep learning. Paper presented at the *International Joint Conference on Computational Intelligence*, 321-327.
- IAEA. (2015). *Plant life management models for long term operation of nuclear power plants*. (). Vienna: INTERNATIONAL ATOMIC ENERGY AGENCY. Retrieved from

<https://www.iaea.org/publications/10520/plant-life-management-models-for-long-term-operation-of-nuclear-power-plants>

Kathuria, A. (2018). What's new in YOLOv3? Retrieved from <https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b>

Lu, Q., Xie, X., Parlikad, A. K., Schooling, J. M., & Konstantinou, E. (2020). Moving from building information models to digital twins for operation and maintenance. *Proceedings of the Institution of Civil Engineers-Smart Infrastructure and Construction*, , 1-11.

Morales, F. (2019). Keras-ocr [computer software]. <https://keras-ocr.readthedocs.io/en/latest/>:

NVIDIA. (2019). NVIDIA CUDA toolkit. Retrieved from <https://developer.nvidia.com/cuda-toolkit>

OpenCV Team. (2021). OpenCV. Retrieved from <https://opencv.org/about/>

Pishdad-Bozorgi, P., Gao, X., Eastman, C., & Self, A. P. (2018). Planning and developing facility management-enabled building information model (FM-enabled BIM). *Automation in Construction*, 87, 22-38.

Python Software Foundation. (2021). Python. Retrieved from <https://www.python.org/about/>

Redmon, J. (2016). *Darknet: Open source neural networks in C*. Unpublished manuscript. Retrieved from <http://pjreddie.com/darknet/>

Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv Preprint arXiv:1804.02767*,

- Stoitchkov, D., Breier, P., Slepicka, M., Genc, C., Harmsen, F., Köhler, T., . . . Borrmann, A. (2019). Automatic detection of plan symbols in railway equipment engineering using a machine learning approach. Paper presented at the *Proceedings of the 2019 European Conference on Computing in Construction, Chania, Crete, Greece*, 92-99.
- Su, L. (2017). Digitalization and application research of BIM-based power plants lifecycle information. Paper presented at the *International Symposium for Intelligent Transportation and Smart City*, 218-224.
- Wang, X., Lu, Y., Wang, D., Liu, L., & Zhou, H. (2017). Using jaccard distance measure for unsupervised activity recognition with smartphone accelerometers. Paper presented at the *Asia-Pacific Web (Apweb) and Web-Age Information Management (Waim) Joint Conference on Web and Big Data*, 74-83.
- Warsop, T., & Singh, S. (2010). A survey of object recognition methods for automatic asset detection in high-definition video. Paper presented at the *2010 IEEE 9th International Conference on Cyberntic Intelligent Systems*, 1-6.
- Xu, J., Lu, W., Xue, F., & Chen, K. (2019). ‘Cognitive facility management’: Definition, system architecture, and example scenario. *Automation in Construction*, 107, 102922.  
doi:<https://doi.org/10.1016/j.autcon.2019.102922>
- Yonghye Kwon. (2019). Yolo\_label.exe. Retrieved from  
[https://github.com/developer0hye/Yolo\\_Label](https://github.com/developer0hye/Yolo_Label)



Yuhan, N., Chimay, A., & Weisheng, L. (2019). Taxonomy and deployment framework for emerging pervasive technologies in construction projects. *Journal of Construction Engineering and Management*, 145(5), 04019028. doi:10.1061/(ASCE)CO.1943-7862.0001653

## Appendix A

# Machine Learning-Based Approaches

Below is the MATLAB script for the SIFT approach.

```
%% TAG DETECTION FROM 2D IMAGES SIFT APPROACH
% Method 1 is using SURF and SIFT

%% CLEAR WORKSPACE AND READ IMAGES
clear all
clc
close all

%First read the images you are searching for in the scene (i.e. target)
target = rgb2gray(imread('s10_train.jfif'));
figure;
imshow(target);
title('Tag of interest')

%Now read the image showing the entire environment
scene = rgb2gray(imread('s10.jfif'));
figure;
imshow(scene);
title('Scan environment')

%% DETERMINE FEATURES IN BOTH TARGET AND SCENE
%Detect the feature points
tpoints = detectSURFFeatures(target);
spoints = detectSURFFeatures(scene);

%Plot feature points on target image
figure;
imshow(target)
title('50 Strongest Feature Points from Target Image')
hold on;
plot(selectStrongest(tpoints,50));

%Plot feature points on entire environment
figure;
imshow(scene);
title('300 Strongest Feature Points from Environment');
hold on;
plot(selectStrongest(spoints,300));

%% FEATURE EXTRACTION AND MATCHING BETWEEN SCENE AND TARGET
%Extract the feature descriptors in both images
[tfeatures, tpoints] = extractFeatures(target,tpoints);
[sfeatures, spoints] = extractFeatures(scene, spoints);

%Match features using their descriptors
tpairs = matchFeatures(tfeatures, sfeatures);

%Display matched features
matchedt = tpoints(tpairs(:,1),:);
matcheds = spoints(tpairs(:,2),:);

%Locate object in the scene using these matched points
[tform, inliert, inliers] = estimateGeometricTransform(matchedt,matcheds,'affine');

figure;
```

```

showMatchedFeatures(target, scene, inliert, inliers, 'montage');
title('Matched Points');

%% SHOW TARGET IN SCENE IMAGE USING A BOUNDING BOX
%Boud the target in overall scene
tpolygon = [1, 1; size(target,2),1;size(target,2),size(target,1);1,size(target,1);1,1];
newtpolygon = transformPointsForward(tform,tpolygon);

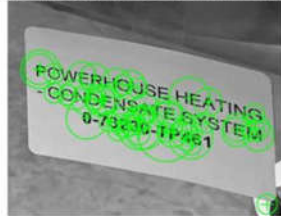
%Display detected object
figure;
imshow(scene);
hold on;
line(newtpolygon(:,1),newtpolygon(:,2),'Color','y','LineWidth',3);
title('Detected Tag');

ocrtxt = ocr(target)

```

The results from the SIFT approach are shown in the image below.

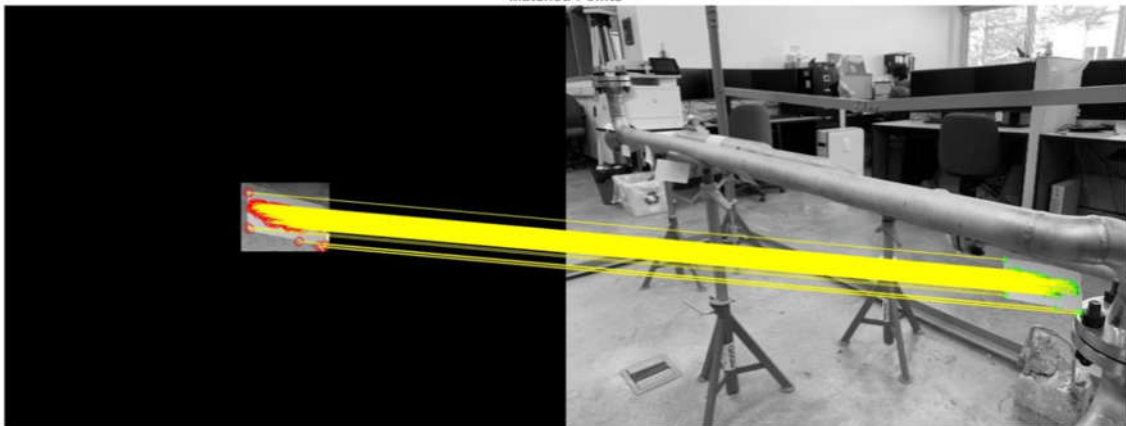
50 Strongest Feature Points from Target Image



300 Strongest Feature Points from Environment



Matched Points



## Below is the MATLAB script for the MSER Approach

```
%% TAG DETECTION FROM 2D IMAGES: MSER Approach
% This method uses MSER regions, filtering with regionprops, then uses OCR

%% CLEAR WORKSPACE
clear all
clc
close all

%% READ IMAGE AND DETECT MSER REGIONS
%Read the image showing the entire environment
colourscene = imread('s2.jfif');
sceneBW = imgaussfilt(rgb2gray(colourscene),2);
scene = histeq(sceneBW);

% Detect MSER regions
[regions, cc] = detectMSERFeatures(scene,'ThresholdDelta',3);

%Visualize MSER regions which are described by pixel lists
figure;
imshow(scene);
hold on
plot(regions,'showPixelList',true,'showEllipses',true);
title('MSER Regions')
hold off

%% SPECIFY REGIONS FOR FILTERING AND THRESHOLDS
mserStats = regionprops(cc,scene,'BoundingBox','Solidity','Image','Area','MaxIntensity');

% Compute the aspect ratio using bounding box data
bbox = vertcat(mserStats.BoundingBox);
w = bbox(:,3);
h = bbox(:,4);
aspectRatio = w./h;

% Threshold data to determine which regions to remove. Custom to image
filter = aspectRatio > 4;
filter = filter | [mserStats.Solidity] < 0.3;
filter = filter | [mserStats.Area] < 0.6*mean('Area');
filter = filter | [mserStats.MaxIntensity] < 1.1*mean('MaxIntensity');

% Remove regions
mserStats(filter) = [];
regions(filter) = [];

% Show remaining regions
figure
imshow(scene)
hold on
plot(regions,'showPixelList',true,'showEllipses',true)
title('After removing filter regions based on geometric properties')
hold off

% Remove non-text regions
regionImage = mserStats(6).Image;
regionImage = padarray(regionImage, [1 1]);

%% PREPROCESSING FOR OCR
distanceImage = bwdist(~regionImage);
skeletonImage = bwmorph(regionImage,'thin',inf);
strokeWidthImage = distanceImage;
strokeWidthImage(~skeletonImage) = 0;
```

```

% Show region image along with stroke width image
figure
subplot(1,2,1)
imagesc(regionImage)
title('Region Image')

subplot(1,2,2)
imagesc(strokeWidthImage)
title('Stroke Width Image')

% Compute stroke width variation metric
strokeWidthValues = distanceImage(skeletonImage);
strokeWidthMetric = std(strokeWidthValues)/mean(strokeWidthValues);

% Threshold the stroke width variation metric
strokeWidthThreshold = 0.5;
strokeWidthFilterIdx = strokeWidthMetric > strokeWidthThreshold;

% Process remaining regions
for j = 1:numel(mserStats)
    regionImage = mserStats(j).Image;
    regionImage = padarray(regionImage, [1 1],0);

    distanceImage = bwdist(~regionImage);
    skeletonImage = bwmorph(regionImage, 'thin',inf);

    strokeWidthValues = distanceImage(skeletonImage);

    strokeWidthMetric = std(strokeWidthValues)/mean(strokeWidthValues);

    strokeWidthFilterIdx(j) = strokeWidthMetric > strokeWidthThreshold;
end

% Remove regions based on the stroke width variation
regions(strokeWidthFilterIdx) = [];
mserStats(strokeWidthFilterIdx) = [];

% Show remaining regions
figure
imshow(scene)
hold on
plot(regions, 'showPixelList', true,'showEllipses',false)
title('After removing non-text regions based on stroke width variation')
hold off

%% BOUNDING BOXES
bboxes = vertcat(mserStats.BoundingBox);

xmin = bboxes(:,1);
ymin = bboxes(:,2);
xmax = xmin + bboxes(:,3) - 1;
ymax = ymin + bboxes(:,4) - 1;

% Expand by a small amount
expAmt = 0.02;
xmin = (1-expAmt)*xmin;
ymin = (1-expAmt)*ymin;
xmax = (1+expAmt)*xmax;
ymax = (1+expAmt)*ymax;

% Clip bounding boxes to be within the image bounds
xmin = max(xmin, 1);
ymin = max(ymin, 1);
xmax = min(xmax, size(scene,2));
ymax = min(ymax, size(scene,1));

```

```

% Show boxes
expBBox = [xmin ymin xmax-xmin+1 ymax-ymin+1];
IexpBBox = insertShape (colourscene, 'Rectangle', expBBox, 'LineWidth', 3);

figure
imshow(IexpBBox)
title('Expanded Bounding Boxes Text')

% Compute overlap ratio
overlapRatio = bboxOverlapRatio(expBBox, expBBox);

% Set overlap ratio between a bounding box and itself to zero to simplify
% graph representation
n = size(overlapRatio,1);
overlapRatio(1:n+1:n^2) = 0;

% Create graph
g = graph(overlapRatio);

% Find the connected text regions within the graph
componentIndices = conncomp(g);

% Merge the boxes based on min and max
xmin = accumarray(componentIndices', xmin, [], @min);
ymin = accumarray(componentIndices', ymin, [], @min);
xmax = accumarray(componentIndices', xmax, [], @max);
ymax = accumarray(componentIndices', ymax, [], @max);

% Compose bounding boxes
textBBoxes = [xmin ymin xmax-xmin+1 ymax-ymin+1];

%% IDENTIFY BOUNDING BOXES WITH TEXT
numRegionsInGroup = histcounts(componentIndices);
textBBoxes(numRegionsInGroup == 1, :) = [];

n = size(textBBoxes);
areaTBB = zeros(n(1),1);
j = n(1);

for i = 1:j
    areaTBB(i) = abs(textBBoxes(i,1)-textBBoxes(i,3))*abs(textBBoxes(i,2)-textBBoxes(i,4));
end

maxATBB = max(areaTBB);
maxthresATBB = 0.2;
minthresATBB = 0.1;

for i = 1:j
    if areaTBB(i) > (1-maxthresATBB)*maxATBB
        textBBoxes(i,:) = [0 0 0 0];
    elseif areaTBB(i) < minthresATBB*maxATBB
        textBBoxes(i,:) = [0 0 0 0];
    end
end

withText = [];
for i = 1:j
    if sum(textBBoxes(i,:)) > 0
        withText = [withText; textBBoxes(i,:)];
    end
end

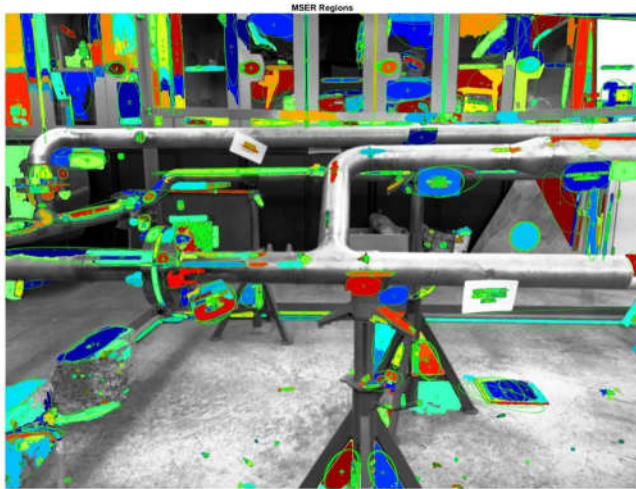
ITextRegion = insertShape(colourscene, 'Rectangle', withText, 'LineWidth', 5);

```

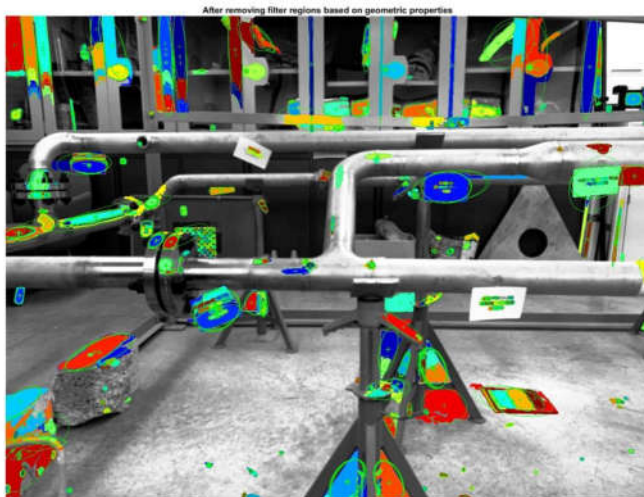
```
%% OUTPUT IMAGE WITH TAG DETECTION
figure
imshow(ITextRegion)
title('Detected')

ocrtxt = ocr(scene, withText);
[ocrtxt.Text]
```

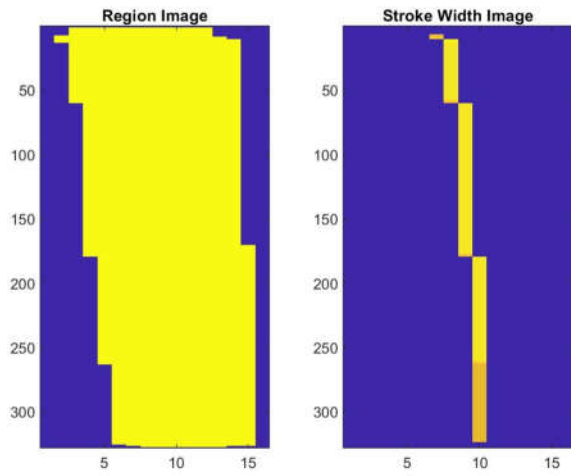
The results from the MSER approach are shown below. These images were the output of the code above.



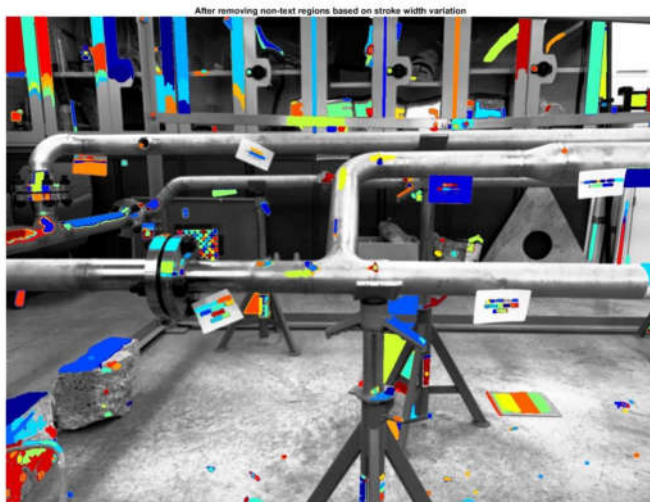
Initial MSER regions



Removing regions based on geometric properties

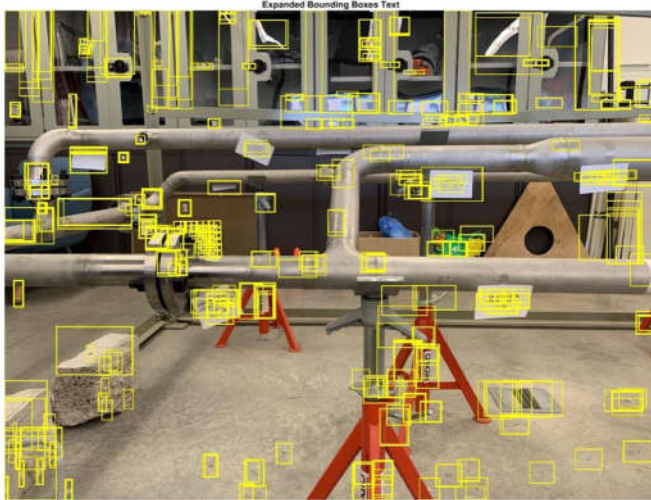


Determining the stroke width which will be used to detect potential text.

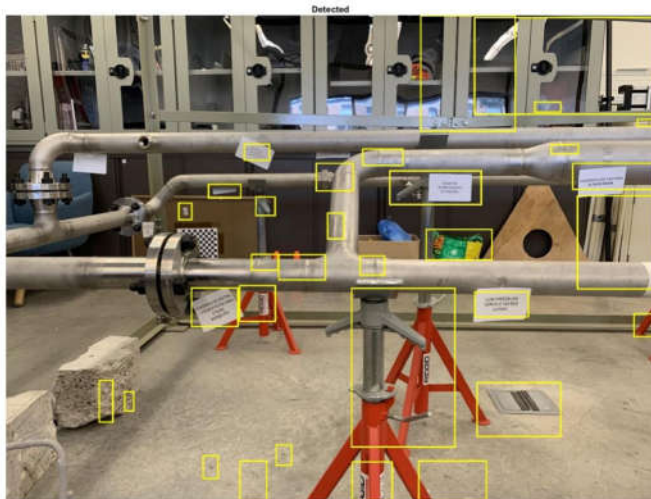


Further filtering of regions based on stroke width.





First round of bounding box detections.



Final round of filtering to reduce number of detections. Various false positives were still included despite final filtering.

Below is the MATLAB script for the Hough Transform approach. This code was written by Cristobal Lara, a collaborator on this experimental testing.

```

%% CLEAR WORKSPACE
clear
close all
fclose('all');

%% PARAMETER DEFINITION
FName = '0-72800-MV19.JPG';

FName = strcat(FolderPath, '\', FName);

%Tag dimensions (Width, Height) (Unitless, it can be any unit)
TagDm0 = [10.2, 6.35]; %Real dimensions of the tags

%Range for computing factor range
%Minimum size in pixels

```

```

MinDm = 40;
%Maximum size (proportion of the picture)
MaxDmF = 0.4; %Value from 0 to 1;
%Delta in pixels between scaling factors
dPxF = 3; %pixels

%Vector for trials of Hough Transform
Thxv = (0:5:60)';
Thyv = (0:5:60)';
Thzv = (-90:5:85)';

% Threshold for detecting gray-color range
%As a coefficient of variation of color
ThdCol = 7.5/100;
ThdBr = 0.6;

%% LOAD VARIABLE AND INITIAL IMAGE PROCESSING
I = imread(FName);
nI = size(I); %Pixel resolution on the image
IG = rgb2gray(I); %Monocolor image conversion

%Compute the highest and lowest brightness
Imean = mean(I,3);
RangBr = [min(min(Imean)),max(max(Imean))];
ThdBr = ThdBr*diff(RangBr)+RangBr(1);

%Compute the Factor size vector
FacVec = [MinDm/min(TagDm0),0.1*max(nI)/max(TagDm0)];
TagDm = repmat(TagDm0,2,1).*repmat(FacVec',1,2);
FacVec = linspace(TagDm(1,1),TagDm(2,1),10)'./TagDm(1,1);
FacVec = unique(round(FacVec,0));
TagDm = TagDm(1,:);

%% LOOP TO COMPUTE ALL HOUGH TRANSFORMS AND LOCATE PEAKS

%Create ID matrix to identify each loop
%Factor, Theta-X, Theta-Y, Theta-Z
IDLoop = allcomb(FacVec,Thxv,Thyv,Thzv);
%Total number of iterations
nIt = size(IDLoop,1);

%Maximum number of peaks to locate initially in the Hough transform
Npk = 100;

Res = repmat(struct('NPeaks',0,'HPks',0,'Magn',0,'ColAvg',0,'ColCov',0),nIt,1);

%thickness for the rectangle contour
t = round(max(0.025*min(TagDm),1),0);

tic
for i=1:nIt
    %% COMPUTE MASK/KERNEL FOR CONVOLUTION
    %Create Mask, filled masked, and relative coordinates for
    %the vertices
    [Mask,MaskF,MaskCorn] = FunctionBox3DProj(TagDm,...
        1,IDLoop(i,2),IDLoop(i,3),IDLoop(i,4),...
        t);
    %Values for normalization
    NMask = sum(sum(Mask))/t; %Number of points in perimeter
    AMaskF = sum(sum(MaskF)); %Area of the mask region
    %Size of the mask
    sMask = size(Mask); %Pixel dimensions on the mask
    of = (sMask-1)/2; %offset to each of the edges of the mask from the
center

```

```

%% Downsample image
    %A downsampling technique is implemented in the image to be
    %processed to speed up the calculations. The intention is
    %to simulate an increase in the kernel.
    Is = I(1:IDLoop(i,1):end,1:IDLoop(i,1):end,:);
    BWs = IG(1:IDLoop(i,1):end,1:IDLoop(i,1):end);
    BWs = imgaussfilt(BWs,3);
    BWs = edge(BWs,'Roberts');
    nIs = size(BWs);

%% OBTAIN HOUGH TRANSFORM (CONVOLUTION)
%Compute the mask convolution (Hough Space)
    HSp = conv2(BWs,Mask,'same');

    MxHSp = max(max(HSp));
    ThdHSp = 0.1;

%Get the highest points in the Hough space
    HPks = houghpeaks(HSp,Npk,'Threshold',ThdHSp*MxHSp);

%% 1st FILTER - DISCARD LOW CONFIDENCE PEAKS
%Transform subindices to index values
    Ind = sub2ind(nIs,HPks(:,1),HPks(:,2));
%Get magnitudes normalized
    MHPks = HSp(Ind)/NMask;
%Filter with threshold
    Val = MHPks>=0.2;
    HPks = HPks(Val,:);
    MHPks = MHPks(Val);
%Update number of peaks gathered
    nPks = size(HPks,1);

%% 2st FILTER - DISCARD POINTS CLOSE TO THE EDGE
%Discard first peaks near the edges
    Valc = or(HPks(:,1)<=of(1),HPks(:,1)+of(1)>=nIs(1));
    Valr = or(HPks(:,2)<=of(2),HPks(:,2)+of(2)>=nIs(2));
    Val = or(Valc,Valr);
    HPks = HPks(~Val,:);
%Update number of peaks gathered
    nPks = size(HPks,1);

%% 3rd FILTER - BY REGION (NOT OVERLAPPING TAGS)
%Eliminate results too close from each other, getting the
%maximum within certain area
    j=1;
    while j<=nPks-1
        %Loop through each peak result
        %Get the location index for the mask
        indr = [HPks(j,1)-of(2),HPks(j,1)+of(2)];
        indc = [HPks(j,2)-of(1),HPks(j,2)+of(1)];
        %Create a binary image of the same size of the image
        %and locat the Mask on it
        IOv1 = false(nIs(1:2));
        IOv1(indr(1):indr(2),indc(1):indc(2)) = MaskF;

        %Loop throug the rest of the results to compare against
        k=j+1;
        while k<=nPks
            %Get the lcoation indexes for the mask for the
            %second peak of comparison
            indr = [HPks(k,1)-of(2),HPks(k,1)+of(2)];
            indc = [HPks(k,2)-of(1),HPks(k,2)+of(1)];
        end

        %Extract from the reference binary image the
        %location of the next peak found
    end

```

```

IOVp = IOv1(indr(1):indr(2),indc(1):indc(2));
%Find the intersection between masks.
IOv2 = and(IOVp,MaskF);

%If there is an intersection, then it means that
%the mask overlap and then the secondary point will
%be discarded
Ov = find(IOv2,1,'first');
if ~isempty(Ov)
    %Secon peak poin discarded
    HPks(k,:)=[];
    nPks = nPks-1; %Updating the number of peaks
else
    %If there is no intersection, proceed to next iteration
    k=k+1;
end
end
%Once the peak has been compared against the rest of
%the peaks, it proceeds to do the same with the next
%point.
j=j+1;
end

%% 4th FILTER - BY BRIGHNESS+COLOR
%Vector to filter the peaks based on their color value
PkVal = false(nPks,1);
AvColV = zeros(nPks,1);
CovColV = zeros(nPks,1);
%Loop to extract color information on the areas found
for j=1:nPks
    %Define coordinates in pixels to extract from the
    %original image (color)
    indr = [HPks(j,1)-of(2),HPks(j,1)+of(2)];
    indc = [HPks(j,2)-of(1),HPks(j,2)+of(1)];
    %Extract portion of the original image
    Ip = Is(indr(1):indr(2),indc(1):indc(2),:);
    %Extract only the window portion
    MaskF3 = repmat(MaskF,1,1,3);
    Ip = double(Ip).*MaskF3;
    Val2 = reshape(MaskF,sMask(1)*sMask(2),1);
    Val3 = reshape(MaskF3,sMask(1)*sMask(2)*3,1);
    Ip(~Val3) = nan;
    %Evaluate color conditions
    AvCol = mean(Ip,3); %Mean color per pixel
    SDCol = std(Ip,1,3); %std color per pixel
    CovCol = SDCol./AvCol; %COV color per pixel
    %Average values (Overall)
    AvColV(j) = sum(sum(AvCol(Val2)))/AMaskF;
    CovColV(j) = sum(sum(CovCol(Val2)))/AMaskF;
    %Evaluate if color is bright enough and if its gray
    if and(AvColV(j)>=ThdBr,CovColV(j)<=ThdCol)
        PkVal(j)=true;
    end
end
%Eliminate false positives based on Color
HPks = HPks(PkVal,:);
nPks = size(HPks,1);
AvColV = AvColV(PkVal);
CovColV = CovColV(PkVal);
%Get magnitude of Hough Space
Ind = sub2ind(nIs,HPks(:,1),HPks(:,2));
MHPks = HSp(Ind)/NMask;

%% SCALE UP THE LOCATIONS FOUND
HPks = (HPks-1)*IDLoop(i,1)+1;

```

```

%% WRAPP UP THE RESULTS OF THE ITERATION INTO A GLOBAL VARIABLE
%Get the magnitudes of the HoughSpace and normalize it with
%the size of the tag
Res(i).NPeaks = nPks;
Res(i).HPks = HPks;
Res(i).Magn = MHPks;
Res(i).ColAvg = AvColV;
Res(i).ColCov = CovColV;

%----- TESTING CODE -----
%Obtain the coordinates for the enclosing box
if nPks>0
    PolyCord = zeros(nPks,4*2);
    PolyCord(:,1:2:end) =
(MaskCorn(:,1)*IDLoop(i,1)+repmat(HPks(:,2)',4,1))';
    PolyCord(:,2:2:end) =
(MaskCorn(:,2)*IDLoop(i,1)+repmat(HPks(:,1)',4,1))';

    txt = strings(nPks,1);
    txt(:,1) = num2str(MHPks,'%0.3f');

    Iann = insertShape(I,'polygon',PolyCord,'LineWidth',8,'Color','r');
    Iann =
insertText(Iann,PolyCord(:,1:2),txt,'FontSize',18,'BoxColor',...
'w','BoxOpacity',0.6,'TextColor','black');

%
    Fig = figure('Visible','off');
    Fig = figure('Visible','on');

    imshow(Iann)
    ax = gca;
    Head = {'IT','F','Thx','Thy','Thz'};
    TitleText = join(strcat(Head,num2str([i;IDLoop(i,:)'],'%02.2f'))','-
');

    ax.Title.String = TitleText;

    FigPath = strcat(SavePath,'\ ',TitleText{1},'.png');
    print(Fig,FigPath,'-dpng','-r600')
end

```

Below is the function to generate the rectangular mask/kernel involved in the Hough Transform approach shown above.

```

function [MatR,MatF,PolyCord] = FunctionBox3DProj(RecDm,F,Thx,Thy,Thz,t,unit)
%% DESCRIPTION
%It returns the kernel of the projection of a rectagle into the X-Y
%plane with dimensions 'X' (width) and 'Y' height, scaled with a factor
%'F' and rotated around X, Y and Z axis in space.
%% INPUT:
%RecDm: Matrix/vector of 2x1 or 1x2 with the Widht Height of
%the rectangle (pixels) [X,Y];
%F: Factor to scale the rectangle size [DEFAULT: 1]
%Thx: Angle of rotation around the 'x' axis [DEFAULT: 0]
%Thy: Angle of rotation around the 'y' axis [DEFAULT: 0]
%Thz: Angle of rotation around the 'z' axis [DEFAULT: 0]
%t: Thickness of the line (in proportion to the minimum
%dimension of the rectangle.[Optional]
%Set up to the minimum of 1 pixel.
%unit: Units for the angles of rotation [DEFAULT: degrees]

%% OUTPUT:
%MatR: Logical matrix (zeros and ones) where the rectangle with
%the projection of the rectangle. The size of the final kernel

```

```

        %it is a square of NxN, with N being twice largest dimension of X
        %and Y plus one.
        %MatF: Logical matrix (zeros and ones) with the same
        %rectangular shape as MatR but filled.
        %ROI: Rectangle object with coordinates referenced to the
        %center of the rectangle within the Mask

%% INPUT HANDLING
%Default size factor
if nargin<=1 || isempty(F)
    F=1;
end
%Default angles of rotation zero
if nargin<=2 || isempty(Thx)
    Thx=0;
end
if nargin<=3 || isempty(Thy)
    Thy=0;
end
if nargin<=4 || isempty(Thz)
    Thz=0;
end
%Check if the angle units is defined (Default is degrees)
if nargin<=6 || isempty(unit)
    unit = 'deg';
else
    %If it was defined, check that it either of the two available
    %options
    %Cut the unit length in case the user wrote the full word
    unit = unit(1:3);
    if ~or(strcmp(unit,'deg'),strcmp(unit,'rad'))
        %if any of the possible units was input, then set the default
        %value
        unit = 'deg';
    end
end
end

%% PROCESS
X =RecDm(1);
Y =RecDm(2);
%Get scaled size of rectangle
nX = ceil(X*F);
nY = ceil(Y*F);
nM1 = max([nX,nY]);
nM = 2*nM1+1;
%Total number of points used for the coordinates
nP = 2*nX+2*nY;

%Plane of projection (vectors)
N = repmat([0,0,1],nP,1); %Normal vector to plane X-Y

%Preallocate output variable
MatR = false(nM);

%Build coordinates for unrotated rectangle
x1=[1:nX]'; %Y' coordinates for horizontal lines
y1=[1:nY]'; %Y' coordinates for vertical lines
%Assemble the points
Cord=[x1, repmat(nY,nX,1), zeros([nX,1]);
      repmat(nX,nY,1), flipud(y1), zeros([nY,1]);
      flipud(x1), ones([nX,1]), zeros([nX,1]);
      ones([nY,1]), y1, zeros([nY,1])];

%Rotate rectangle coordinates and project them into X-Y plane

```

```

Cord = RotXYZ(Thx,Thy,Thz,unit)*Cord';
Cord = Cord';
Cord = cross(N,cross(Cord,N));

%Center the cordinates around the origin and transform into pixels
Cord = Cord(:,1:2)-mean(Cord(:,1:2));
Cord = round(Cord+nM1+1);

%Translate coordinates into matricial indices for picture (Note that
%the rows and columns are switched, and the row number are flip so the
%pixel count start from the top as it is for pictures
CordInd = sub2ind([nM,nM],Cord(:,2),nM-Cord(:,1));

%Assign true values to the output boolean matrix
MatR(CordInd) = true;

%If the filled matrix is requested
if nargin>=2
    MatF = imfill(MatR,'holes');
    %To show the kernel image:
    imshow(MatF);
end

%Increase thickness of line
if ~or(nargin<=5,isempty(t))
    %Compute number of pixels
    t = max(round(t,0),1);
    %Check if the additional thickness is larger than 1
    if t>1
        t=t-1;
        MatRT = false(nM+2*t);
        for i=1:t*2+1
            MatRT(i:nM+i-1,2*t+1:nM+2*t) = MatRT(i:nM+i-1,2*t+1:nM+2*t)+MatR;
            MatRT(2*t+1:nM+2*t,i:nM+i-1) = MatRT(2*t+1:nM+2*t,i:nM+i-1)+MatR;
        end
        %Clip extra borders
        MatR = MatRT(2*t+1:nM+2*t,2*t+1:nM+2*t);
    end
end

if nargin>=3
    % ROI = images.roi.Polygon;
    % x0 = Cord(1,1)-nM1-1;
    % y0 = Cord(1,2)-nM1-1;
    % w = round(sqrt(sum(diff(Cord([1,nX],:)).^2)));
    % h = round(sqrt(sum(diff(Cord([nX+1,nX+nY],:)).^2)));
    % if diff(Cord([1,nX],1))==0
    %     th = pi()/2;
    % else
    %     th = atan(diff(Cord([1,nX],2))/diff(Cord([1,nX],1)));
    % end
    % PolyCord = Cord([1,nX,nX+nY,nX*2+nY],:)-nM1-1;
    % PolyCord = [PolyCord(:,1),-PolyCord(:,2)];
end

%To show the kernel image:
imshow(MatR);

end

%% NESTED FUNCTIONS
function RotM = RotXYZ(Thx,Thy,Thz,unit)
%% DESCRIPTION
%This functions compute a rotational matrix around the 'x', 'y', and 'z'
%axis.

```

```

%% INPUT:
%Thx: Rotation angle around 'X' axis.
    %Angle starts at 'y' axis, and towards the positive 'z'
    %axis counterclockwise.
%Thy: Rotation angle around 'Y' axis.
    %Angle starts at 'z' axis, and towards the positive 'x'
    %axis counterclockwise.
%Thz: Rotation angle around 'Z' axis.
    %Angle starts at 'x' axis, and towards the positive 'y'
    %axis counterclockwise.
%unit: Units for the input angles
    %It needs to be a character vector
    %Either 'deg' or 'rad' for degrees and radians
    %respectively.
    %[OPTIONAL: Default value is 'deg' (degrees)]

%% OUTPUT:
%RotM: 3D rotational matrix that accounts the three angles of
    %rotation. The matrix has a dimension of 3x3.

%% INPUT HANDLER
%Check if the angle units is defined
if nargin<=3 || isempty(unit)
    unit = 'deg';
else
    %If it was defined, check that it either of the two available
    %options
    %Cut the unit length in case the user wrote the full word
    unit = unit(1:3);
    if ~or(strcmp(unit,'deg'),strcmp(unit,'rad'))
        %if any of the possible units was input, then set the default
        %value
        unit = 'deg';
    end
end

%% FUNCTION PROCEDURE
%Change the angles into a vector
Th = [Thx,Thy,Thz];
%Convert the angle to radians (if applicable)
if strcmp(unit,'deg')
    Th = Th.*pi()./180;
end
%Call subfunctions to compute the rotational matrix for each axis
Rtx=rotx(Th(1));    Rty=roty(Th(2));    Rtz=rotz(Th(3));

%Compute the global rotational matrix
RotM = Rtz*Rty*Rtx;
end

%Rotational matrix around X axis
function Rt = rotx(Th)
%% DESCRIPTION
%It computes the rotational matrix around 'x' axis.
%% INPUT:
%Th: Rotational angle in radians
    %Angle starts at 'y' axis, and towards the positive 'z'
    %axis counterclockwise.
%% OUTPUT:
%Rt: 3D Rotational matrix for the 'x' axis (3x3 matrix).

%% FUNCTION PROCEDURE
Rt = [1,0,0;
    0, cos(Th), -sin(Th);
    0,sin(Th),cos(Th)];
end

```



```

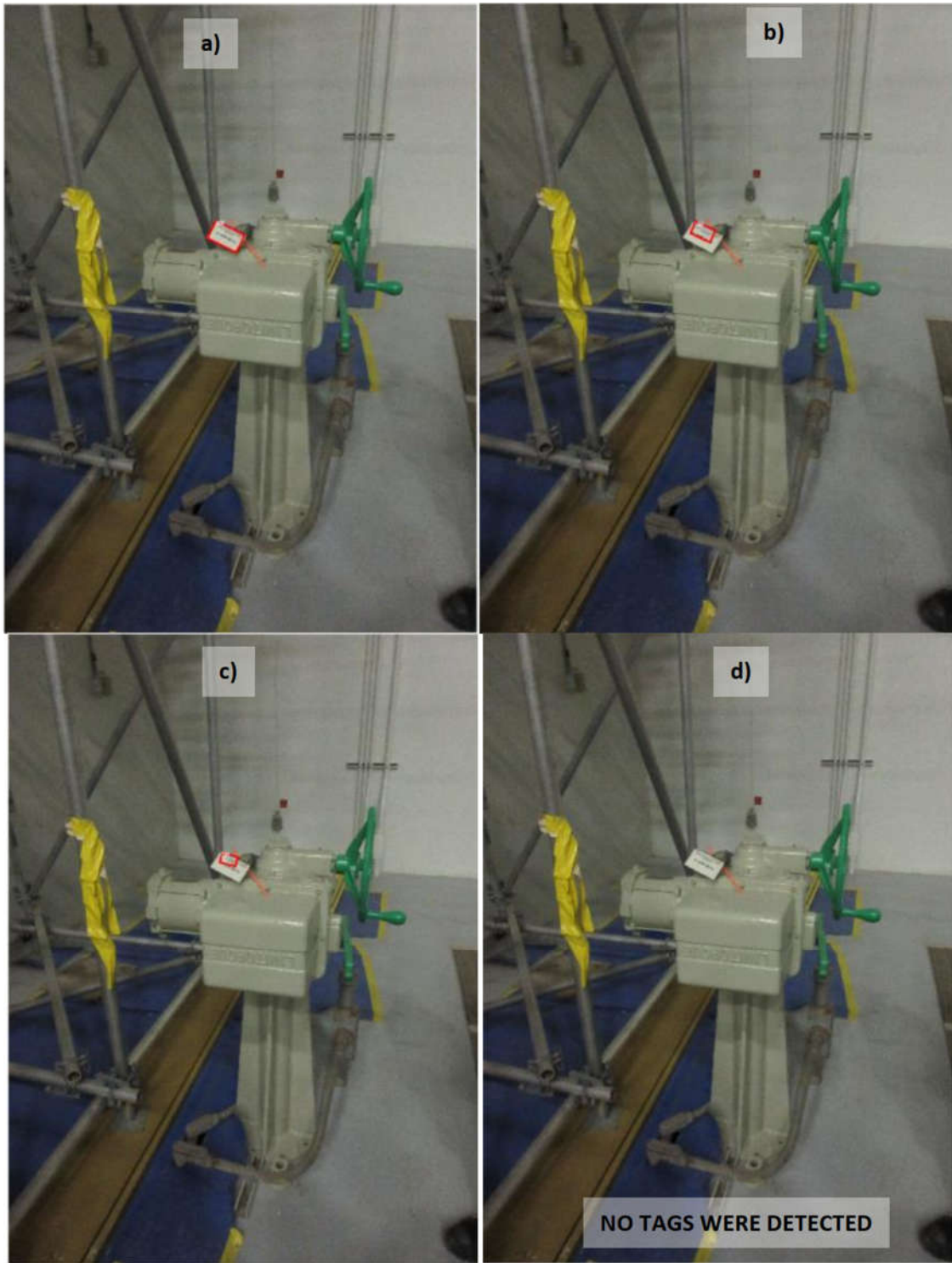
% Rotational matrix around Y axis
function Rt = roty(Th)
%% DESCRIPTION
%It computes the rotational matrix around 'y' axis.
%% INPUT:
    %Th: Rotational angle in radians
        %Angle starts at 'z' axis, and towards the positive 'x'
        %axis counterclockwise.
%% OUTPUT:
    %Rt: 3D Rotational matrix for the 'y' axis (3x3 matrix).

%% FUNCTION PROCEDURE
    Rt = [cos(Th), 0, sin(Th);
          0, 1, 0;
          -sin(Th), 0, cos(Th)];
end
% Rotational matrix around Z axis
function Rt = rotz(Th)
%% DESCRIPTION
%It computes the rotational matrix around 'z' axis.
%% INPUT:
    %Th: Rotational angle in radians
        %Angle starts at 'x' axis, and towards the positive 'y'
        %axis counterclockwise.
%% OUTPUT:
    %Rt: 3D Rotational matrix for the 'z' axis (3x3 matrix).

%% FUNCTION PROCEDURE
    Rt = [cos(Th), -sin(Th), 0;
          sin(Th), cos(Th), 0;
          0, 0, 1];
end

```

Results of the Hough Transform approach are shown below. Each image represents a different tag size and rotation angle. A) exact tag size and rotation angle  $F_1 = 1$ ,  $\theta_z = -26.5^\circ$  B)  $F_1 = 0.66$ ,  $\theta_z = -25^\circ$  C)  $F_1 = 0.5$ ,  $\theta_z = -15^\circ$  and D)  $F_1 = 1.5$ ,  $\theta_z = -15^\circ$



Below is the MATLAB script for the White Regions approach. This script was developed by Cristobal Lara, a collaborator on the experimental test setup.

```

%% CLEAR WORKSPACE
clear
close all
fclose('all');

%% PARAMETERS
%SELECT FOLDERS TO READ AND SAV
%Folder where the pictures are located
PathFolder = 'D:\UWaterloo\Courses\CIVE 700 - Infrastructure Computer
Vision\Project\Matlab\Images';
%Folder where the process pictures with the boundaries highlighted
%will be placed
SaveFolder = 'D:\UWaterloo\Courses\CIVE 700 - Infrastructure Computer
Vision\Project\Matlab\ImageswBoxes';
%Select if the ouput is desired
%Show figure?
ShwFig = false;
%Save Figure?
SvFig = true;

%Parameters
GsSigma = 3; %Sigma for the gaussian filter
ThdCol = 5/100; %Theshold for covariance to isolate whites
ThdBr = 0.65; %Threshold for brightness of the color;
%Threshold for the minimum dimension on a tag
MinDimP = 2/100; %Percentage relative to the size of the picture
MinDimD = 40; %Minimum size in pizel
%The code will choose the largest of the above two
%Threshold for the maximum size of the tag
MaxDimP = 0.5; %Percentage relative to the size of the picture
%Factor (INTEGER) to donwsample the image
F=1;
%Tag Dimension (Units are not important, the ratio is the important
%part. (FFirst dimension is the horizontal, second the vertical).
TagDm0 = [10.2,6.35];
%Maximum angle of rotation to compute different aspect ratios
Thxv = [0,60]';
Thyv = [0,60]';
Thzv=0;

%% PROCESS
%Preprocess variables before starting the loop
%Compute ratio
TagDm0 = TagDm0/TagDm0(2);
%Principal axis of the tag
TagDmLamb = [TagDm0(1),0,0;0,TagDm0(2),0];
%Compute the aspect ratios expected based on 3D tag rotation
[EigV, ID] = FunComputeVRot(TagDmLamb,Thxv,Thyv,Thzv);
%Get all the filenames to be opened
[FilesN] = FunGetFileNames(PathFolder,{'JPG','jpg'});
%Number of files to open
nF = size(FilesN,1);
for kk=1:nF
%% LOAD IMAGE
I = imread(FilesN{kk,1});
I = I(1:F:end,1:F:end,:); %Downsample image
%Image size
nI = size(I);
IG = rgb2gray(I); %Grayscale image
Ib = imgaussfilt(I,GsSigma); %Blured image (Color)

```

```

%% COLOR STATISTICS OF IMAGE
%Transform the blurred color image to double matrix
    Id = double(Ib);
%Statistics per pixel
AvCol = mean(Id,3); %Mean color per pixel (Brighthness)
SDCol = std(Id,1,3); %std color per pixel
CovCol = SDCol./AvCol; %COV color per pixel

%Only leave image that is white enough
%First redefine threshold based on image levels
    RangBr = [min(min(AvCol)),max(max(AvCol))];
    ThdBrL = ThdBr*diff(RangBr)+RangBr(1);
%Filter hte color image based on the threshold
%Validation vectors (boolean matrix)
    WhiteVal = and(AvCol>=ThdBrL,CovCol <=ThdCol); %For 2D
    WhiteVal3 = reshape(repmat(WhiteVal,1,1,3),prod(nI),1); %For 3D (Color
matrix)

%Binarize the image in black and white
    Iw2 = false(nI(1:2));
    Iw2(reshape(WhiteVal,prod(nI(1:2)),1)) = true;

%% FIND BOUNDARIES IN BINARY IMAGE
%Find Boundaries, label matrix and number of regions
[Bnd,L,nRg] = bwboundaries(Iw2,'noholes');
%Compute the statistics of the regions found
stats = regionprops(L,'Area','Centroid','BoundingBox',...
    'SubarrayIdx','PixelIdxList','ConvexHull',...
    'MajorAxisLength','MinorAxisLength','Orientation');

%% 1ST FILTER - BY SIZE
%Discard elements too small
    MinAxL = [stats.MinorAxisLength]';
    MinDim = max(MinDimP*min(nI(1:2)),MinDimD);
    Val = MinAxL<MinDim;
    stats(Val) = [];
    Bnd(Val) = [];
    nRg = size(stats,1);
%Discard elements too large
    MaxAxL = [stats.MajorAxisLength]';
    MaxDim = MaxDimP*max(nI(1:2));
    Val = MaxAxL>MaxDim;
    stats(Val) = [];
    Bnd(Val) = [];
    nRg = size(stats,1);

%% 2nd FILTER - By Aspect Ratio
%Verify the slender ratio with the possible rotations of the
%expected tag size
    %Minimum and maximum ratio for the eigenvectors
    minRt = min(EigV(:,end)); mxRt = max(EigV(:,end));
    %Exclude all the ratios that do not fall into these range
    lmb1 = [stats.MinorAxisLength]';    lmb2 = [stats.MajorAxisLength]';
    %Ratios
    Rt = lmb2./lmb1;
    %Eliminate ratios greater than the ones expected
    Val = or(Rt>mxRt,Rt<minRt);
    stats(Val) = [];
    Bnd(Val) = [];
    %Number of remaining regions
    nRg = size(stats,1);

%% 3rd FILTER - by Shape
% Verify if the the point form a rectangle shape
%Allocate error variables
    Err = zeros(nRg,1); ErrN = zeros(nRg,7);

```

```

%Loop through each region
for i=1:nRg
    %Minor and Major principal axis (Not used at the
    %moment)
    MinAxL = stats(i).MinorAxisLength;
    MajAxL = stats(i).MajorAxisLength;
    %Orientation of the major principal axis
    Th = stats(i).Orientation;
    %Get boundaries for each region
    B = Bnd{i};
    B = [-B(:,2),B(:,1)];
    %Compute the error with the enveloping box for the
    %points selected
    [Err(i),ErrN(i,:)] = RectMeasErr(MinAxL,MajAxL,Th,B);
end
%Discard regions with errors too large
Val = ErrN(:,end)>15/100;
stats(Val) = [];
Bnd(Val) = [];
Err(Val)=[];
ErrN(Val,:)=[];
nRg=size(stats,1);

%% SAVE PICTURE
%Create figure
if ShwFig
    Fig = figure('Visible','on');
else
    Fig = figure('Visible','off');
end
%Create string vector with label number for each regions
txt = strings(nRg,1);
txt(:,1) = num2str((1:nRg)','%02.0f');
%Get the centroid of each region
Cent = zeros([nRg,2]);
ind = sub2ind([nRg,2],reshape(repmat(1:nRg,2,1),1,2*nRg),...
             repmat(1:2,1,nRg));
Cent(ind) = [stats.Centroid];
%Create annotated figure with the labels for each region
if nRg > 0
    Iann = insertText(I,Cent,txt,'FontSize',30,'BoxColor',...
                    'w','BoxOpacity',0.6,'TextColor','black');
else
    Iann = I;
end
%Plot/Display the annotated figure
imshow(Iann)
hold on
%Plot the boundary regions
t = max(0.001*min(nI(1:2)),3);
for k=1:nRg
    B = Bnd{k};
    plot(B(:,2), B(:,1), 'r', 'LineWidth', 3)
end
%Save figure
if SvFig
    Fname = strcat(SaveFolder,'\ ',FilesN{kk,2});
    print(Fig,Fname,'-dpng','-r600')
end
%Close figures opened before the next cycle
close all
end

```

Below is the function to find the images in the specified folder, which was involved in the White Regions approach shown above.

```
function [FilesN] = FunGetFileNames(Path,ExtO)
%% DESCRIPTION
%The function will get all the files names in the indicated folder that
%has the right file extension
%% INPUT
%Path: Path for the folder
%ExtO:
%% OUTPUT
%FilesN Cell variable with two columns on it. The first column
%has the complete path for the file, and the second the
%name of the file and the third column is the
%date of each file.
%% INPUT HANDLING
if nargin<=1 || isempty(ExtO)
    ExtO = {'.txt','.csv'};
end
%% PROCEDURE
%Get all files in the folder
listing = dir(Path);
listing = struct2table(listing);
nFt = size(listing,1);
%Isolate only the files with the proper extension
nExt = length(ExtO);
ValExt = false(nFt,1);
for i=1:nExt
    ValExtT = cellfun(@ExtFind,listing.name,repmat(ExtO(i),nFt,1));
    ValExt = or(ValExt,ValExtT);
end
%number of files to be opened
nF = sum(ValExt);
%Return the file names
if nF>=1
    FilesN(1:nF,1) = strcat(Path,'\',listing.name(ValExt));
    FilesN(1:nF,2) = listing.name(ValExt);
    FilesN(1:nF,3) = listing.date(ValExt);
end
end

%% NESTED FUNCTIONS
function ValExt = ExtFind(Txt,ExtO)
%% DESCRIPTION
%The function finds if the filename has the desired extension name
%% INPUT
%Txt: Filename
%ExtO: Extension for the file including the point, example
%".txt"
%Optional value, the default extensions are .txt and
%.csv
%% OUTPUT
%ValExt: Boolean vector indicating if the file has the right
%extension or not
%% INPUT HANDLING
if nargin<=1 || isempty(ExtO)
    ExtO = '.txt';
end
%% PROCEDURE
%Find locations of all points in file name
ind = strfind(Txt,'.');
%Assign the extension to the filename
```

```

if isempty(ind)
    ExtT = '.';
else
    ind = ind(end);
    ExtT = Txt(ind:end);
end
%Default value to not open the file
ValExt = false;
%Validate for the desired extension
if strcmp(ExtT,ExtO)
    ValExt = true;
end
end
end

```

Below is the function to compute the principal axis for the varying tag view angles and projections.

```

function [EigV, ID] = FunComputeVRot (V, Thxv, Thyv, Thzv, unit)
%% DESCRIPTION
%The function computes the length of the principal axis of the
%rectangular tag with vector dimension V, rotated and projected into
%the plane 'X-Y'.
%% INPUT:
    %V: Vector with principal axis of the rectangle to be rotated
    %EXAMPLE:
        %V = [dx,0,0;0,dy,0];
        %where dx and dy are the dimension in x and dy.
    %Thxv: Vector with all the angles of rotation around 'X' axis
    %EXAMPLE:
        %Thxv = (0:5:60)';
        %In this example the angles are in degrees. If the
        %user desires to use rad, specify the variable unit
    %Thyv: Idem but rotation around 'Y' axis
    %Thyz: Idem but rotation around 'Z' axis
    %unit: Units for the input angles
        %It needs to be a character vector
        %Either 'deg' or 'rad' for degrees and radians
        %respectively.
        %[OPTIONAL: Default value is 'deg' (degrees)]
%% OUTPUT
    %EigV: Eigen vector matrix. The numeric matrix columns are
    %configured as follow:
        %1) Magnitude of the projection of vector 'dx'
        %2) Magnitude of the projection of vector 'dy'
        %3) Angle of rotation for dx with respect of the
        %horizontal plane
        %4) Angle of rotation for dy with respect of the
        %horizontal plane
        %5) Ratio between projections of dx/dy
    %ID: Matrix that cross reference the different combination
    %tested for the angles of rotation with the respective
    %row in the EigV
%% INPUT HANDLING
    %Check if the angle units is defined
    if nargin<=4 || isempty(unit)
        unit = 'deg';
    else
        %If it was defined, check that it either of the two available
        %options
        %Cut the unit length in case the user wrote the full word
        unit = unit(1:3);
        if ~or(strcmp(unit,'deg'),strcmp(unit,'rad'))
            %if any of the possible units was input, then set the default
            %value
            unit = 'deg';
        end
    end

```

```

end
end

%% PROCEDURE
%Create matrix with all rotation combinations
ID = allcomb(Thxv,Thyv,Thzv);
%Total number of combinations
nCmb = size(ID,1);
%Size of vectors to be tested (It supposed to be only two, but
%in case the user has extra dimensions
nV = size(V,1);
%Preallocate the output variable of Eigenvalues (principal
%axis)
EigV = zeros(nCmb,2*nV+1);
%Normal vector to the plane of projection
N = repmat([0,0,1],nV,1);

% Loop to test each combination
for i=1:nCmb
    %Rotate the vector in the 3D space
    v1 = RotXYZ(ID(i,1),ID(i,2),ID(i,3),unit)*V';
    %Compute the projection of the vector in the plane
    v2 = cross(N,cross(v1',N)); %The result matrix show a vector for each row

    %Computes the magntiude for the the vector
    EigV(i,1:nV) = sqrt(sum(v2.^2,2))';

    %Compute the angle of rotation for each vector
    for j=1:nV
        if v2(1,1) == 0
            EigV(i,nV+j) = 90*sign(v2(j,2));
        else
            EigV(i,nV+j) = rad2deg(atan(v2(j,2)/v2(j,1)));
        end
    end
end

%Compute the ratio between the frist two magntiudes dy/dx
EigV(:,2*nV+1) = EigV(:,1)./EigV(:,2);
end

%% NESTED FUNCTIONS
function RotM = RotXYZ(Thx,Thy,Thz,unit)
%% DESCRIPTION
%This functions compute a rotational matrix around the 'x', 'y', and 'z'
%axis.
%% INPUT:
    %Thx: Rotation angle around 'X' axis.
        %Angle starts at 'y' axis, and towards the positive 'z'
        %axis counterclockwise.
    %Thy: Rotation angle around 'Y' axis.
        %Angle starts at 'z' axis, and towards the positive 'x'
        %axis counterclockwise.
    %Thz: Rotation angle around 'Z' axis.
        %Angle starts at 'x' axis, and towards the positive 'y'
        %axis counterclockwise.
    %unit: Units for the input angles
        %It needs to be a character vector
        %Either 'deg' or 'rad' for degrees and radians
        %respectively.
        %[OPTIONAL: Default value is 'deg' (degrees)]
%% OUTPUT:
    %RotM: 3D rotational matrix that accounts the three angles of
    %rotation. Thhe matrix has a dimension of 3x3.

```



```

%% INPUT HANDLER
%Check if the angle units is defined
if nargin<=3 || isempty(unit)
    unit = 'deg';
else
    %If it was defined, check that it either of the two available
    %options
    %Cut the unit length in case the user wrote the full word
    unit = unit(1:3);
    if ~or(strcmp(unit,'deg'),strcmp(unit,'rad'))
        %if any of the possible units was input, then set the default
        %value
        unit = 'deg';
    end
end

%% FUNCTION PROCEDURE
%Change the angles into a vector
Th = [Thx,Thy,Thz];
%Convert the angle to radians (if applicable)
if strcmp(unit,'deg')
    Th = Th.*pi()./180;
end

%Call subfunctions to compute the rotational matrix for each axis
Rtx=rotx(Th(1));    Rty=roty(Th(2));    Rtz=rotz(Th(3));

%Compute the global rotational matrix
RotM = Rtz*Rty*Rtx;
end

%% NESTED FUNCTIONS
%Rotational matrix around X axis
function Rt = rotx(Th)
%% DESCRIPTION
%It computes the rotational matrix around 'x' axis.
%% INPUT:
    %Th: Rotational angle in radians
        %Angle starts at 'y' axis, and towards the positive 'z'
        %axis counterclockwise.
%% OUTPUT:
    %Rt: 3D Rotational matrix for the 'x' axis (3x3 matrix).

%% FUNCTION PROCEDURE
Rt = [1,0,0;
      0, cos(Th), -sin(Th);
      0,sin(Th),cos(Th)];
end
% Rotational matrix around Y axis
function Rt = roty(Th)
%% DESCRIPTION
%It computes the rotational matrix around 'y' axis.
%% INPUT:
    %Th: Rotational angle in radians
        %Angle starts at 'z' axis, and towards the positive 'x'
        %axis counterclockwise.
%% OUTPUT:
    %Rt: 3D Rotational matrix for the 'y' axis (3x3 matrix).

%% FUNCTION PROCEDURE
Rt = [cos(Th),0,sin(Th);
      0, 1, 0;
      -sin(Th),0,cos(Th)];
end
% Rotational matrix around Z axis
function Rt = rotz(Th)

```

```

%% DESCRIPTION
%It computes the rotational matrix around 'z' axis.
%% INPUT:
    %Th: Rotational angle in radians
        %Angle starts at 'x' axis, and towards the positive 'y'
        %axis counterclockwise.
%% OUTPUT:
    %Rt: 3D Rotational matrix for the 'z' axis (3x3 matrix).

%% FUNCTION PROCEDURE
    Rt = [cos(Th), -sin(Th), 0;
          sin(Th), cos(Th), 0;
          0, 0, 1];
end

```

Below is the function to measure the region border points.

```

function [Err,ErrN] = RectMeasErr(MinAxL,MajAxL,Th,P)
%% DESCRIPTION
%Function to compute the error of the region border points "P" with respect
%of an envelope rectangle
%% INPUT
    %MinAxL: Minor principal axis length of the border region points
    %MaxAxL: Major principal axis length of the border region points
    %Th: Angle for the major principal axis
    %P: Two column matrix with the coordinates of the border
        %region points.
%% OUTPUT
    %Err: Average error, equal to the sum of the distances of the
        %points to the closest point in the envelope rectangle
        %divided by the number of points in the region border
    %ErrN: Normalized error with respect of the length of the envelope
        %rectangle

%% PROCEDURE
    %Rotate and move to the origin the data set
    %Add z'coordinate to data point set
    nP = size(P,1);
    P = [P,zeros(nP,1)];
    %Rotate the data point set and move it to the origin
    [U,~,~] = svd(cov(P));
    R=inv(U);
    P = (R*P)'; %Rotate
    P = P-mean(P);
    %Translate to the origin
    P(:,1) = P(:,1)-(max(P(:,1))+min(P(:,1)))/2;
    P(:,2) = P(:,2)-(max(P(:,2))+min(P(:,2)))/2;
    %Round the dimensions for the rectangle
    nX= round(max(P(:,1))-min(P(:,1)),0);
    nY= round(max(P(:,2))-min(P(:,2)),0);
    %Build the theoretical rectangle
    x1=[1:nX]'; %Y' coordinates for horizontal lines
    y1=[1:nY]'; %Y' coordinates for vertical lines
    %Assemble the points
    Cord=[x1, repmat(nY,nX,1), zeros([nX,1]);
          repmat(nX,nY,1), flipud(y1), zeros([nY,1]);
          flipud(x1), ones([nX,1]), zeros([nX,1]);
          ones([nY,1], y1, zeros([nY,1])];
    %Center them to the origin
    Cord = Cord-mean(Cord);

    %Find distances to nearest point

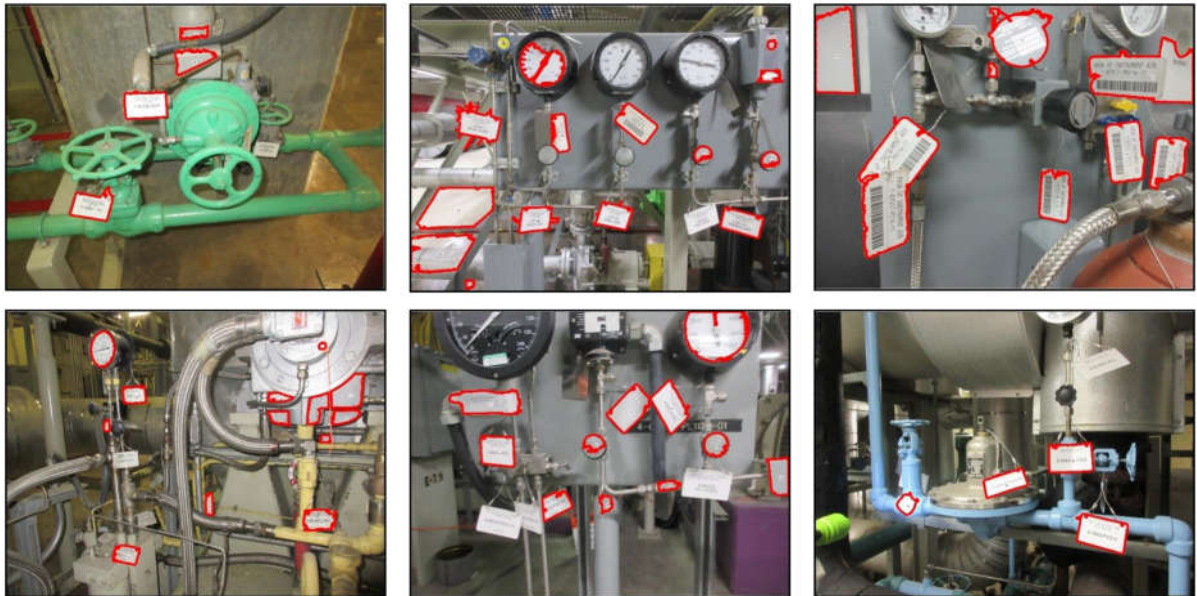
```

```

[~,D] = knnsearch(Cord,P);
%Error is the sum of all the distances (pixels)
Err = sum(D);
%The normalize error is the average distance divided by the minimum
%side length of the theoretical rectangle
sigm = std(D);
avg = mean(D);
ErrN = [avg,sigm,[nX,nY],sigm/avg,avg/min(nX,nY),sigm/min(nX,nY)];
%%-----CODE DEBUGGING-----
%
%   figure()
%   plot(Cord(:,1),Cord(:,2));
%   hold on
%   plot(P(:,1),P(:,2));
%%-----CODE DEBUGGING-----
end

```

The White Regions approach was evaluated on multiple sample images as shown below. The red lining was used as a bounding box for the detected objects.



## Appendix B

### Deep Learning-Based Approach

Below is the Python code that was used to run the Auto-linking pipeline.

```
# -*- coding: utf-8 -*-
"""
Auto-linking
This code was developed as part of research to detect and read
equipment ID tags within an NPP facility.
The object detection was trained on a custom dataset.
"""

# import the necessary packages that will be used in the code
import os
import re
import subprocess
from coordsmapping import CoordsMapping
import keras_ocr
from PIL import ImageGrab, Image
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import shutil

# read files that will be used in the auto-linking analysis. Test.txt
# contains file paths to the necessary files
recog = keras_ocr.recognition.Recognizer()
from subprocess import check_output
afile = open("autolinking_test/converted_scans/test.txt", "r")
files = afile.readlines()
pipeline = keras_ocr.pipeline.Pipeline()

# set a destination folder for the auto-linking output images and files
destination = "darknet/autolinking_test/detections"

def changename(file_name):
    end = file_name.rfind('\n')
    start = file_name.rfind('\\')
    return file_name[start+1:end]

def get_digits(str1):
    num_digits = 0
    for i in str1:
        if i.isdigit():
            num_digits += 1
    return num_digits

def getcoords(text):
```

```

coords = []
tags = []
idx = 0

# this while loop will obtain the bounding box coordinates from the output
file
while (text.find('(',idx)>0):
    tagidx = text.find('OPG_',idx)
    coloidx = text.find(':',tagidx)
    tags.append(text[tagidx:coloidx])
    open_para = idx = text.find('(', idx)
    close_para = idx = text.find(')', idx)
    coords.append(text[open_para+1:close_para])
return coords[1:], tags

total_data = []

# the for loop runs the detection, image cropping, ocr, and data
collection for each image
for file_name in files[0:24]:
    data = []
    new_name = changename(file_name)
    test_name = "test_"+new_name
    conv_name = new_name[:-4]
    print(conv_name)
    print("MY NAME", new_name)
    if(file_name.find(' ')!=-1):
        continue
    print("start")
    # ./darknet.exe detector test data/OPGobj_data/obj.data
    cfg/yolov3_4.cfg data/OPGobj_data/backup/yolov3_last.weights
    tests/test2.JPG -i 0 -thresh 0.5 -dont-show
    command = "./darknet.exe detector test obj.data cfg/yolov3_4.cfg
    data/OPGobj_data/backup/yolov3_2_14.weights
    autolinking_test/converted_scans/"+str(new_name)+" -i 0 -thresh 0.01 -
    dont show -ext_output"
    commands = []
    print(command)
    output = subprocess.call(["./darknet", "detector", "test", "obj.data",
    "cfg/yolov3_4.cfg", "data/OPGobj_data/backup/yolov3_2_14.weights", "autolink
    ing_test/converted_scans/"+str(new_name), "-i 0", "-thresh 0.01", "-dont-
    show", "-ext_output"], stdout=open(str(new_name)[:-4]+".txt", "w"))
    print("rename")
    try:
        os.rename("predictions.jpg", new_name)
    except:
        print("renaming"+new_name)
    try:
        shutil.move(new_name, "autolinking_test/detections/"+new_name)
        shutil.move(new_name[:-4]+".txt",
    "autolinking_test/detections/"+new_name[:-4]+".txt")
    except:

```

```

    print("moving"+new_name)
print("end")
print("OCR")
preds = open("autolinking_test/detections/" + new_name[:-4]+".txt")
print("autolinking_test/detections/" + new_name[:-4]+".txt")
text = str(preds.readlines())
coords, tags = getcoords(text)
print(coords, tags)

images = []
for i, coordinates in enumerate(coords):
    asplit = coordinates.split('\t')
    s = ''.join(asplit)
    nums = list(map(int, ''.join([x if x.isdigit() else ' ' for x in
s]).split()))
    print(nums)
    img = Image.open("autolinking_test/converted_scans/"+new_name)
    #img.show()
    im = img.crop((nums[0], nums[1], nums[0]+nums[2],
nums[1]+nums[3]))
    pixelref = int(nums[0]+nums[2]/2), int(nums[1]+nums[3]/2)
    print(pixelref)
    nc = CoordsMapping(conv_name, pixelref)
    newcoords = nc.pntint_conv()
    print(newcoords)
    im.save('autolinking_test/cropped/crop_'+str(i)+new_name)
    numpy_img = [np.array(im)]
    try:
        prediction_groups = pipeline.recognize(numpy_img)
    except:
        continue
    aprediction = ''
    print(prediction_groups[0])
    for prediction in prediction_groups[0]:
        num_digits = get_digits(prediction[0])
        if (len(prediction[0])>len(aprediction)):
            if num_digits> 2:
                data =
[new_name, prediction[0], tags[i], prediction[0], prediction[0], 'hotlink',
newcoords[0], newcoords[1], newcoords[2], 'visiByids', 'Scans', 'originalids']
                print(prediction[0])
                aprediction = prediction[0]

    img.close()
    total_data.append(data)
print(coords, "\n \n")

# the data is exported in .CSV format, including equipment ID information
that is used for linking
total_data = [x for x in total_data if x != []]
print(total_data)
columns =
"image", "name", "category", "label", "link", "type", "posx", "posy", "posz", "visi

```

```

Byids", "visiBynames", "originalids"
all_data = np.array(total_data)
df = pd.DataFrame(data = all_data, columns= columns)
print(df.head())
df.to_csv('autolinking_test/output_autolinking.csv')

```

Below is the CoordsMapping Class that computes the point coordinates that correspond with the object detection bounding box.

```

from PIL import Image
import xmltodict
import numpy as np
import cv2

class CoordsMapping:
    def __init__(self, filename, detcoords):
        self.newfilename = filename.replace(" ", "_")
        self.rgbfile = "autolinking_test/converted_scans/"+filename+".jpg"
        self.xyzfile = "autolinking_test/converted_scans/"+filename+".png"
        self.xmlfile = "autolinking_test/converted_scans/"+filename+".xml"
        self.detcoords = detcoords
        self.openfiles()
        self.clean_data()

    def openfiles(self):

        rgbimage = Image.open(self.rgbfile)
        self.rgbw, self.rgbh= rgbimage.size

        xyzimage = Image.open(self.xyzfile)
        self.xyzmatrix = cv2.imread(self.xyzfile,cv2.IMREAD_UNCHANGED)

        self.xyzw, self.yzlh= xyzimage.size

        with open(self.xmlfile) as xml:
            self.params = xml.readlines()[1]

    def clean_data(self):

        paramsdict = xmltodict.parse('<?xml version="1.0"
?><' + self.newfilename + '>' + self.params + '</' + self.newfilename + '>')

        self.scale = float(paramsdict[self.newfilename]['scale'])
        self.offsetx = float(paramsdict[self.newfilename]['offset']['x'])
        self.offsety = float(paramsdict[self.newfilename]['offset']['y'])
        self.offsetz = float(paramsdict[self.newfilename]['offset']['z'])
        self.offset = [self.offsetx, self.offsety, self.offsetz]

    def pntint_conv(self):

```



```

row = int(self.detcoords[0]/self.rgbw*self.xyzw)
col = int(self.detcoords[1]/self.rgbh*self.xyzh)
pt_integer = self.xyzmatrix[row][col]
reverse = np.array(pt_integer[::-1])
product = reverse*self.scale
pt_world = product + self.offset
return pt_world

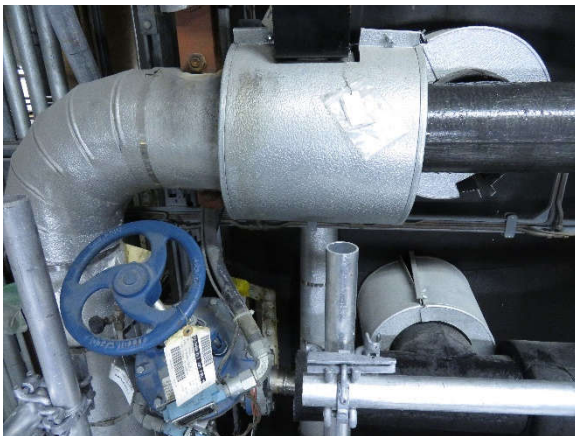
```

```

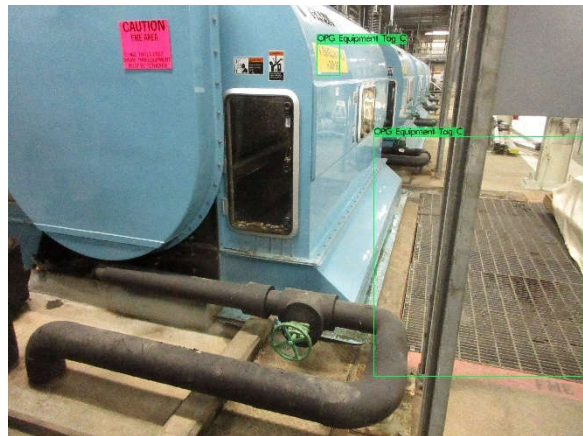
newcoords = nc.pntint_conv()
print(newcoords)

```

Below are a set of example output images demonstrating the detection of the Equipment ID tags. Included are the image names and number of true positives (TP), false positives (FP), and false negatives (FN) corresponding to each example.



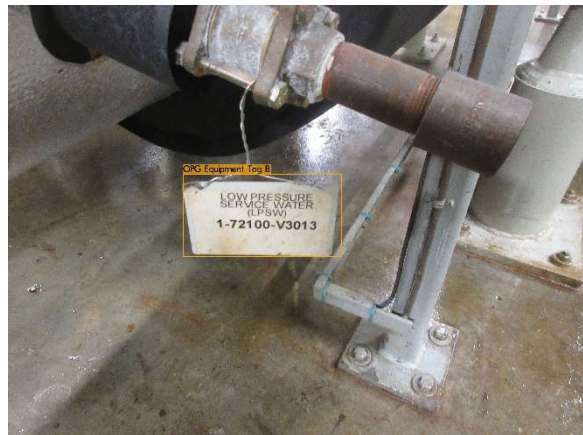
IMG\_1056.JPG, FN:1



IMG\_1129.JPG, TP:1, FP:1

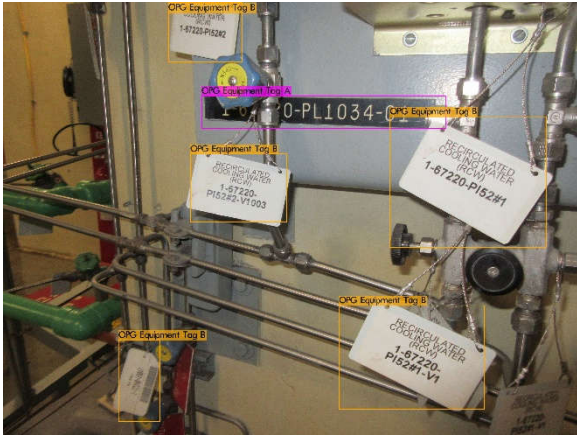


IMG\_1207.JPG, TP:4, FN:1



IMG\_1212.JPG, TP:1

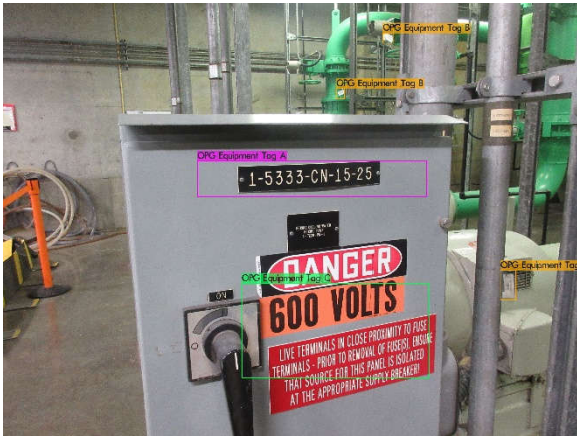




IMG\_1217.JPG, TP:6, FN:1



IMG\_1224.JPG, TP:1



IMG\_1229.JPG, TP:4, FP:1



IMG\_1234.JPG, TP:1



IMG\_1239.JPG, TP:1



IMG\_1244.JPG, FP:1, FN:1



IMG\_1249.JPG, TP:2



IMG\_1254.JPG, TP:4



IMG\_1259.JPG, FN:1



IMG\_1264.JPG, FN:1



IMG\_1274.JPG, TP:2



IMG\_1279.JPG, TP:4, FN:1





IMG\_1284.JPG, FN:2



IMG\_1289.JPG, FN:1



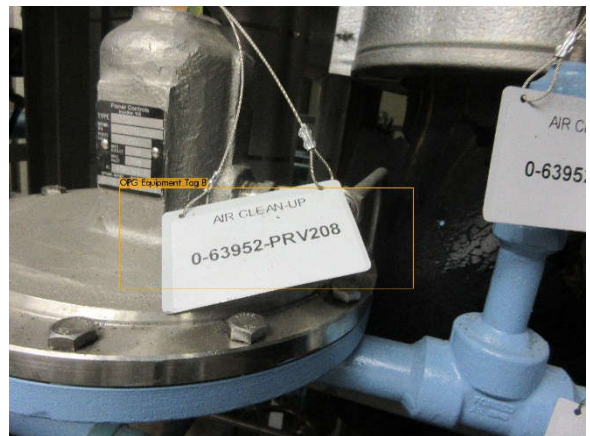
IMG\_1389.JPG, FN:1



IMG\_1423.JPG, FN:1



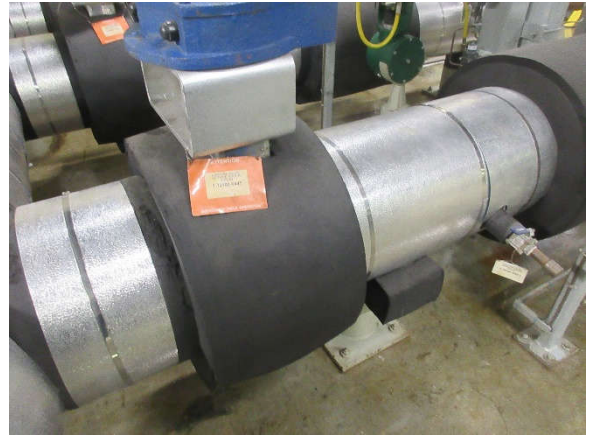
IMG\_1476.JPG, TP:1, FN:1



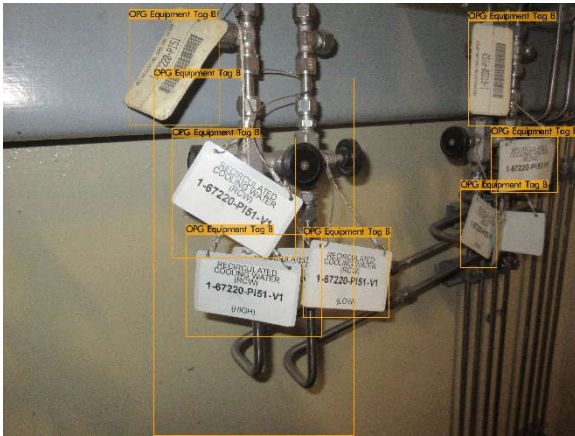
IMG\_2741.JPG, TP:1, FN:1



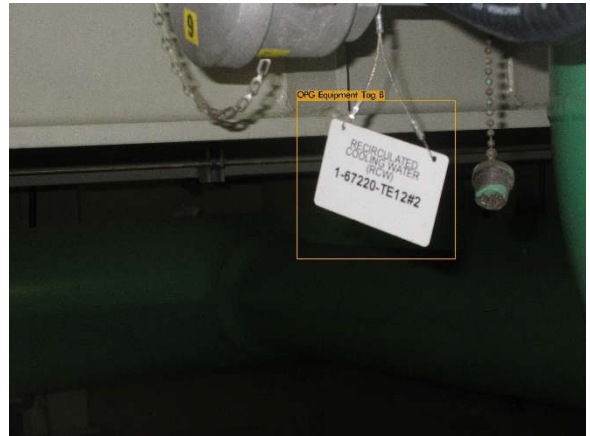
IMG\_1205.JPG, TP:1, FN:1



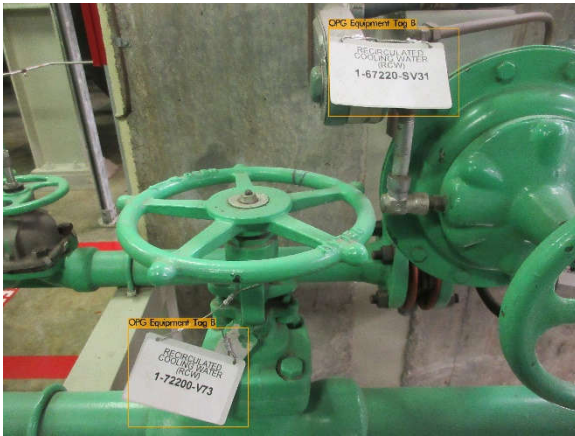
IMG\_1210.JPG, FN:3



IMG\_1215.JPG, TP:7



IMG\_1222.JPG, TP:1



IMG\_1227.JPG, TP:2



IMG\_1232.JPG, TP:1





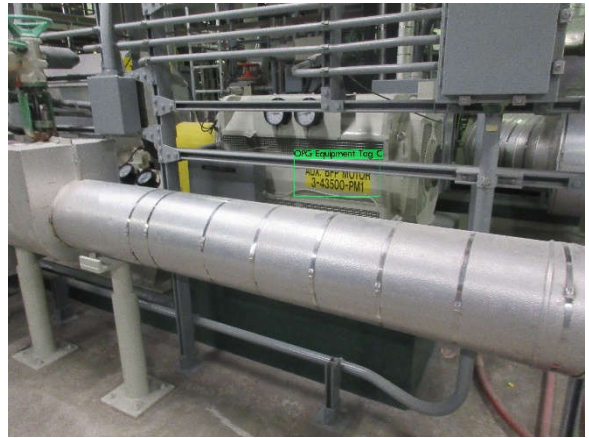
IMG\_1237.JPG, TP:5



IMG\_1242.JPG, FN:1



IMG\_1247.JPG, FN:3



IMG\_1252.JPG, TP:1



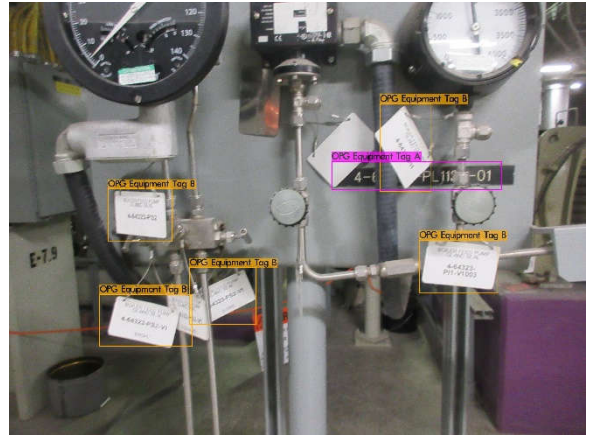
IMG\_1257.JPG, TP:1



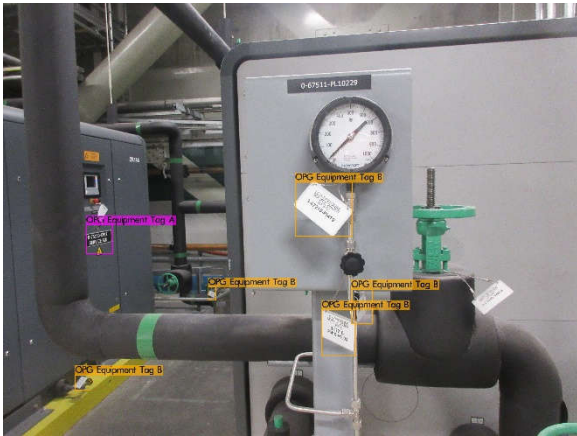
IMG\_1262.JPG, TP:8, FN:3



IMG\_1272.JPG, TP: 2



IMG\_1277.JPG, TP:6, FN:2



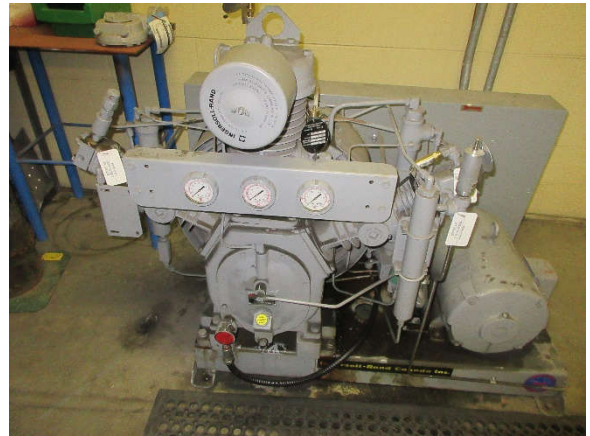
IMG\_1282.JPG, TP:6, FN:2



IMG\_1287.JPG, FP:1, FN:1



IMG\_1292.JPG, TP:1



IMG\_1292.JPG, FN:2





IMG\_1419.JPG, FP:1, FN:1



IMG\_1286.JPG, TP:1

The full list of images, results, and list of possible reasons for the false negative detections is included in the table below. These values were used to conduct the sensitivity analysis and create the confusion matrix.

Image Name	TP	FP	FN	Reason for FN
IMG_1005.JPG			1	Poor resolution
IMG_1009.JPG			2	Poor resolution
IMG_1044.JPG	1			
IMG_1056.JPG			1	Damage
IMG_1067.JPG	1	2		
IMG_1070.JPG			1	Damage
IMG_1122.JPG	2	1		
IMG_1127.JPG	2			
IMG_1129.JPG	1	1		
IMG_1155.JPG	1		1	Covered
IMG_1168.JPG			1	Damage
IMG_1205.JPG	1		1	Normal
IMG_1206.JPG	1			
IMG_1207.JPG	4		1	Covered
IMG_1208.JPG	1	1	1	Normal
IMG_1209.JPG	1			
IMG_1210.JPG			3	Poor resolution
IMG_1211.JPG			1	Poor resolution

Image Name	TP	FP	FN	Reason for FN
IMG_1268.JPG				
IMG_1269.JPG				
IMG_1270.JPG	5			
IMG_1271.JPG	1	1		
IMG_1272.JPG	2	1		
IMG_1273.JPG	4		1	Damage
IMG_1274.JPG	2			
IMG_1275.JPG	1			
IMG_1276.JPG	1			
IMG_1277.JPG	6		2	Covered
IMG_1278.JPG	1			
IMG_1279.JPG	4		1	Covered
IMG_1280.JPG			1	Covered
IMG_1281.JPG	1		2	Normal
IMG_1282.JPG	5		2	Normal
IMG_1283.JPG	2	1	2	Covered
IMG_1284.JPG			2	Poor resolution
IMG_1285.JPG	1			

IMG_1212.JPG	1			
IMG_1213.JPG		1	1	Lighting
IMG_1214.JPG	1			
IMG_1215.JPG	7			
IMG_1216.JPG	5		2	Covered
IMG_1217.JPG	6		1	Covered
IMG_1220.JPG	1			
IMG_1221.JPG	1			
IMG_1222.JPG	1			
IMG_1223.JPG	2			
IMG_1224.JPG	1			
IMG_1225.JPG	1			
IMG_1226.JPG	3			
IMG_1227.JPG	2			
IMG_1228.JPG	1		1	Lighting
IMG_1229.JPG	4	1		
IMG_1230.JPG	1	3		
IMG_1231.JPG	4			
IMG_1232.JPG	1			
IMG_1233.JPG	1			
IMG_1234.JPG	1			
IMG_1235.JPG	1			
IMG_1236.JPG	2		2	Poor resolution
IMG_1237.JPG	5			
IMG_1238.JPG	2			
IMG_1239.JPG	1			
IMG_1240.JPG	3	1		
IMG_1241.JPG	6		1	Normal
IMG_1242.JPG			1	Damage
IMG_1243.JPG			1	Lighting
IMG_1244.JPG		1	1	Lighting
IMG_1245.JPG	6			
IMG_1246.JPG	2			
IMG_1247.JPG			3	Lighting
IMG_1248.JPG	1			
IMG_1249.JPG	2			
IMG_1250.JPG	1			
IMG_1251.JPG			1	Lighting
IMG_1252.JPG	1			

IMG_1286.JPG	1			
IMG_1287.JPG		1	1	Lighting
IMG_1288.JPG			1	Lighting
IMG_1289.JPG			1	Poor resolution
IMG_1290.JPG	10		1	Covered
IMG_1291.JPG	1			
IMG_1292.JPG	1			
IMG_1387.JPG	1		1	Lighting
IMG_1389.JPG			1	Poor resolution
IMG_1390.JPG	2		2	Covered
IMG_1391.JPG	2			
IMG_1393.JPG			2	Lighting
IMG_1396.JPG	2			
IMG_1408.JPG			1	Lighting
IMG_1416.JPG	1			
IMG_1418.JPG	1			
IMG_1419.JPG		1	1	Damage
IMG_1420.JPG	1			
IMG_1423.JPG			1	Damage
IMG_1424.JPG			1	Damage
IMG_1429.JPG			1	Damage
IMG_1431.JPG			1	Damage
IMG_1440.JPG	1			
IMG_1447.JPG	1			
IMG_1448.JPG	1			
IMG_1449.JPG	1			
IMG_1450.JPG	1			
IMG_1452.JPG			2	Lighting
IMG_1476.JPG	1		1	Lighting
IMG_1478.JPG			4	Lighting
IMG_1479.JPG	1		4	Lighting
IMG_1482.JPG			1	Lighting
IMG_1483.JPG	1			
IMG_1502.JPG	1		1	Covered
IMG_1503.JPG			3	Normal
IMG_1504.JPG			1	Normal
IMG_1508.JPG	1		2	Normal
IMG_2210.JPG	1			
IMG_2215.JPG	1			



IMG_1253.JPG	1		
IMG_1254.JPG	4		
IMG_1255.JPG		1	Damage
IMG_1256.JPG	2		
IMG_1257.JPG	1		
IMG_1258.JPG		2	Lighting
IMG_1259.JPG		1	Normal
IMG_1260.JPG	1		
IMG_1261.JPG	2		
IMG_1262.JPG	8	4	Covered
IMG_1263.JPG	5		
IMG_1264.JPG		1	Lighting
IMG_1265.JPG	2	1	Poor resolution
IMG_1266.JPG	1		
IMG_1267.JPG	1		

IMG_2216.JPG			1	Poor resolution
IMG_2337.JPG	2	1		
IMG_2338.JPG	3		2	Poor resolution
IMG_2339.JPG	2		2	Covered
IMG_2340.JPG	1		2	Covered
IMG_2341.JPG	1			
IMG_2460.JPG		1	1	Lighting
IMG_2467.JPG			1	Damage
IMG_2720.JPG	1			
IMG_2721.JPG	1	1		
IMG_2722.JPG	1		1	Poor resolution
IMG_2727.JPG	1			
IMG_2729.JPG			2	Covered
IMG_2740.JPG	4	1		
IMG_2741.JPG	1		1	Covered
Reheater.PNG	1		1	Lighting

## Appendix C

### Database of Existing Equipment ID Tags in Test Dataset

File Path and Image Identifier	No. Visible Tags	Equipment ID (if legible)	Class	Equipment ID (if legible)
C:\Repos\darknet\data\OPGobj_data\IMG_1005.JPG	1		B	
C:\Repos\darknet\data\OPGobj_data\IMG_1009.JPG	2	0-67516-v2316	B B	n/a
C:\Repos\darknet\data\OPGobj_data\IMG_1044.JPG	1	4-47100-tk1	C	condensate storage tank
C:\Repos\darknet\data\OPGobj_data\IMG_1056.JPG	1	0-67516-v2317	B	common instrument air sys
C:\Repos\darknet\data\OPGobj_data\IMG_1067.JPG	1	0-6742-pv195	B	anion exchangers
C:\Repos\darknet\data\OPGobj_data\IMG_1070.JPG	1	0-67516-v2317	B	common instrument air sys
C:\Repos\darknet\data\OPGobj_data\IMG_1122.JPG	2	1-7110-sc2	C B	C/W travelling screen
C:\Repos\darknet\data\OPGobj_data\IMG_1127.JPG	2	1-67150-pl2081-11 7111-sc2	A A	travelling screen local control
C:\Repos\darknet\data\OPGobj_data\IMG_1129.JPG	1	1-7110-sc2	C	C/W travelling screen
C:\Repos\darknet\data\OPGobj_data\IMG_1155.JPG	2	40-RV1 RV1	B C	steam generator steam relief SRV
C:\Repos\darknet\data\OPGobj_data\IMG_1168.JPG	1	3-4XX00-MV2	B	gland seal system
C:\Repos\darknet\data\OPGobj_data\IMG_1205.JPG	2	2-72200-rv138 2-72200-rv133	B B	recirculated cooling water (RCW) recirculated cooling water (RCW)
C:\Repos\darknet\data\OPGobj_data\IMG_1206.JPG	2	2-72200-v134	B B	recirculated cooling water (RCW)
C:\Repos\darknet\data\OPGobj_data\IMG_1207.JPG	5	2-722	B B B	

		2-72200-v132	B	recirculated cooling water (RCW)
		2-72200-v137	B	recirculated cooling water (RCW)
C:\Repos\darknet\data\OPGobj_data\IMG_1208.JPG	2	2-67220-tw151	B	recirculated cooling water (RCW)
		2-67220-tw152	B	recirculated cooling water (RCW)
C:\Repos\darknet\data\OPGobj_data\IMG_1209.JPG	1	1-72200-rv138	B	recirculated cooling water (RCW)
C:\Repos\darknet\data\OPGobj_data\IMG_1210.JPG	3		B	
			B	
		1-72100-v447	B	low pressure service water (LPSW)
C:\Repos\darknet\data\OPGobj_data\IMG_1211.JPG	1		B	
C:\Repos\darknet\data\OPGobj_data\IMG_1212.JPG	1	1-72100-v3013	B	low pressure service water (LPSW)
C:\Repos\darknet\data\OPGobj_data\IMG_1213.JPG	1	1-67210-sv31-#1	B	low pressure service water (LPSW)
C:\Repos\darknet\data\OPGobj_data\IMG_1214.JPG	1	1-67517-prv31-4	B	back up instrument air
C:\Repos\darknet\data\OPGobj_data\IMG_1215.JPG	8			
		1-67220-pi51	B	recirculating cooling water
		1-67220-pi51-v1	B	recirculated cooling water (RCW)
		1-67220-pi51-v1	B	recirculated cooling water (RCW)
		1-67220-pi51-v1	B	recirculated cooling water (RCW)
		1-67220-pi53-v1	B	recirculated cooling water (RCW)
		1-67220-pi53-v1	B	recirculated cooling water (RCW)
		1-67220-pi53	B	recirculating cooling water
C:\Repos\darknet\data\OPGobj_data\IMG_1216.JPG	8			
		1-XX220-PL1034	A	
		1-67220-pi52#2	B	cooling water (RCW)
		1-67220-pi52#2-v1003	B	recirculated cooling water (RCW)
		1-67220-pi52#1-v1	B	recirculated cooling water (RCW)

		1-72200-v9047	B	RCW
		1-67220-pi52#1-v1	B	recirculated cooling water (RCW) (LOW)
		1-67220-pi52#1	B	recirculated cooling water (RCW)
C:\Repos\darknet\data\OPGobj_data\IMG_1217.JPG	9			
		1-67220-pi52#1-v1	B	recirculated cooling water (RCW)
		1-6XX0-PL1034-01	A	
		1-67220-pi52#2	B	recirculated cooling water (RCW)
		1-67220-pi52#2-v1003	B	recirculated cooling water (RCW)
		1-67220-pi52#1	B	recirculated cooling water (RCW)
		1-67220-pi52#1-v1	B	recirculated cooling water (RCW)
		1-72200-v9047	B	RCW
C:\Repos\darknet\data\OPGobj_data\IMG_1220.JPG	1	1-72200-hx2	C	RCW HX
C:\Repos\darknet\data\OPGobj_data\IMG_1221.JPG	1	1-72200-hx2	C	RCW HX
C:\Repos\darknet\data\OPGobj_data\IMG_1222.JPG	1	1-67220-te12#2	B	recirculated cooling water (RCW)
C:\Repos\darknet\data\OPGobj_data\IMG_1223.JPG	2		B	
		1-67220-te12#2	B	recirculated cooling water (RCW)
C:\Repos\darknet\data\OPGobj_data\IMG_1224.JPG	1	1-72200-p1	B	recirc cooling water
C:\Repos\darknet\data\OPGobj_data\IMG_1225.JPG	1	1-72200-pm1	B	recirc cooling water
C:\Repos\darknet\data\OPGobj_data\IMG_1226.JPG	4	1-72200-v72	B	recirculated cooling water (RCW)
		1-67220-sv31	B	recirculated cooling water (RCW)
		1-72200-v72	B	recirculated cooling water (RCW)
		1-72200-pvX6	B	recirculated cooling water (RCW)
C:\Repos\darknet\data\OPGobj_data\IMG_1227.JPG	2	1-67220-sv31	B	recirculated cooling water (RCW)
		1-72200-v73	B	recirculated cooling water (RCW)
C:\Repos\darknet\data\OPGobj_data\IMG_1228.JPG	2	1-72200-v72	B	recirculated cooling water (RCW)
		1-72200-pv66	B	recirculated cooling water (RCW)

C:\Repos\darknet\data\OPGobj_data\IMG_1229.JPG	5			
		1-5333-cn-15-25	A	
		1-7220-pm-1	A	recirc cooling water recirc pump
C:\Repos\darknet\data\OPGobj_data\IMG_1230.JPG	2	1-5333-cn-16-25	A	
		1-7220-pm-2	A	recirc cooling water recirc pump
C:\Repos\darknet\data\OPGobj_data\IMG_1231.JPG	4		B	
			B	
		1-67220-pi21	B	recirculated cooling water (RCW)
		1-72200-v204	B	recirculated cooling water (RCW)
C:\Repos\darknet\data\OPGobj_data\IMG_1232.JPG	4		B	
			B	
			B	
		1-72200-tk2	C	RCW drain tank
C:\Repos\darknet\data\OPGobj_data\IMG_1233.JPG	1	1-72300-v195	B	powerhouse upper level service water (PULSW)
C:\Repos\darknet\data\OPGobj_data\IMG_1234.JPG	1	1-72300-pv46	B	powerhouse upper level service water (PULSW)
C:\Repos\darknet\data\OPGobj_data\IMG_1235.JPG	1	1-72300-pv46	B	powerhouse upper level service water (PULSW)
C:\Repos\darknet\data\OPGobj_data\IMG_1236.JPG	4		B	
			B	
		1-67220-v9019	B	recirculated cooling water (RCW)
		1-67220-v9026	B	recirculated cooling water (RCW)
C:\Repos\darknet\data\OPGobj_data\IMG_1237.JPG	5		B	
			B	
		1-67220-v9026	B	recirculated cooling water (RCW)
		1-67220-v9019	B	recirculated cooling water (RCW)

		1-67220-v9024	B	recirculated cooling water (RCW)
C:\Repos\darknet\data\OPGobj_data\IMG_1238.JPG	2	1-65720-jb1462-21	A	
		1-67220-pl3444-01	A	recirc clg water system
C:\Repos\darknet\data\OPGobj_data\IMG_1239.JPG	1	1-67220-pl3444-01	A	recirc clg water system
C:\Repos\darknet\data\OPGobj_data\IMG_1240.JPG	3		B	
			B	
		1-73230-tk1	C	htg. Stm. Cndste. Tank
C:\Repos\darknet\data\OPGobj_data\IMG_1241.JPG		1-67323-pi35-v1007	B	powerhouse heating condensate system
		1-67323-pi34-v1003	B	powerhouse heating condensate system
		1-67323-pi34	B	htg stm condensate
		1-67323-pi35-v1003	B	powerhouse heating condensate system
		1-67323-pi35	B	htg stm condensate
		1-67323-pi6-v1003	B	powerhouse heating condensate system
		1-67323-ps32-v1003	B	powerhouse heating condensate system
C:\Repos\darknet\data\OPGobj_data\IMG_1242.JPG	1	1-67323	B	powerhouse condensate sys
C:\Repos\darknet\data\OPGobj_data\IMG_1243.JPG	1	1-78110-fhc48	B	fire prot water supply
C:\Repos\darknet\data\OPGobj_data\IMG_1244.JPG	1			
C:\Repos\darknet\data\OPGobj_data\IMG_1245.JPG	7			
		1-67517-pi14-11	B	back up instrument air
		1-67517-prv14-11	B	back up instrument air
		1-67517-rv14-11	B	back up instrument air
		1-67517-prv14-12	B	back up instrument air
		1-67517-v14-11	B	back up instrument air
		1-67517-pi14-12	B	back up instrument air

C:\Repos\darknet\data\OPGobj_data\IMG_1246.JPG	2	1-67517-rb14-12	B	back up instrument air
		1-67517-tk14-11	B	back up instrument air
C:\Repos\darknet\data\OPGobj_data\IMG_1247.JPG	3	1-67517-prv14-13	B	back up instrument air
		1-67517-tk14-12	B	back up instrument air
		1-67517-rv14-13	B	back up instrument air
C:\Repos\darknet\data\OPGobj_data\IMG_1248.JPG	1	1-63862-pl10084-11	A	
C:\Repos\darknet\data\OPGobj_data\IMG_1249.JPG	2	1-67517-prv14-24	B	back up instrument air
		1-67517-prv14-23	B	back up instrument air
C:\Repos\darknet\data\OPGobj_data\IMG_1250.JPG	1	1-63862-rm2	A	d20 in h2O leak detection gamma monitor
C:\Repos\darknet\data\OPGobj_data\IMG_1251.JPG	1	0-84000-v9001	B	heating steam
C:\Repos\darknet\data\OPGobj_data\IMG_1252.JPG	1	3-43500-pm1	C	aux. bfp. Motor
C:\Repos\darknet\data\OPGobj_data\IMG_1253.JPG	1	3-44030-mv67	B	main condensate circuit
C:\Repos\darknet\data\OPGobj_data\IMG_1254.JPG	4		C	
			B	
			B	
		3-43110-hx5a	C	hp feedheater
		3-43110-hx5a	C	hp feedheater
C:\Repos\darknet\data\OPGobj_data\IMG_1255.JPG	1	3-4110-hx5a	C	hp feedheater
C:\Repos\darknet\data\OPGobj_data\IMG_1256.JPG	2	3-64184-lcv49#1	B	first reheater
		3-64184	B	1st stage reheater drains
C:\Repos\darknet\data\OPGobj_data\IMG_1257.JPG	1	3-64184-lcv49#1-v9000	B	first stage reheater drains
C:\Repos\darknet\data\OPGobj_data\IMG_1258.JPG	2	3-41840-v147	B	first stage reheater drains
		3-41840-v146	B	first stage reheater drains
C:\Repos\darknet\data\OPGobj_data\IMG_1259.JPG	1	3-43110-hx5a	C	hp feedheater
C:\Repos\darknet\data\OPGobj_data\IMG_1260.JPG	1	3-67322-jb12670-21	A	
C:\Repos\darknet\data\OPGobj_data\IMG_1261.JPG	1	3-43410-pm1	C	main bfp motor

C:\Repos\darknet\data\OPGobj_data\IMG_1262.JPG	12	3-67210-fs	B	low pressure service water (LPSW)
		3-67210-fs153-v1	B	low pressure service water (LPSW) (HIGH)
		210-fs153-v1		low pressure service water (LPSW) (LOW)
		3-67210-fs151-v1	B	low pressure service water (LPSW) (LOW)
		3-67210-fs	B	
		3-67210-fs152-v1	B	low pressure service water (LPSW) (LOW)
		s152-v1	B	low pressure service water (LPSW) (LOW)
		3-67210-fs152-v4	B	low pressure service water (LPSW) (LOW)
		2-67210-fs153-v5	B	low pressure service water (LPSW) (LOW)
		210-fs153-v1	B	low pressure service water (LPSW) (EQUALIZE)
C:\Repos\darknet\data\OPGobj_data\IMG_1263.JPG	5	3-43410-PM4	C	main bfp motor
		3-43410-pm3	B	
		3-43410-pm3	C	main bfp motor
			B	main bfp motor
C:\Repos\darknet\data\OPGobj_data\IMG_1264.JPG	1	3-64610-pcv102	C	cond. Stm. Disch. Valve
C:\Repos\darknet\data\OPGobj_data\IMG_1265.JPG	3		B	
		3-6410-PCV302	C	cond. Stm. Disch. Valve
C:\Repos\darknet\data\OPGobj_data\IMG_1266.JPG	1	3-48500-tk1	C	atmos. Drain tank



C:\Repos\darknet\data\OPGobj_data\IMG_1267.JPG	2	4-5534-27bio	A	pressurizer htr. 4-3333-htr5
C:\Repos\darknet\data\OPGobj_data\IMG_1268.JPG				
C:\Repos\darknet\data\OPGobj_data\IMG_1269.JPG				
C:\Repos\darknet\data\OPGobj_data\IMG_1270.JPG	5	XXXt301-v1	B	
		4-63617-ft304-v1	B	steam generator level control
		4-63617-ft304-v1	B	steam generator level control
		4-63617-ft304-v5	B	steam generator level control
		4-63617-ft	B	steam generator level control
C:\Repos\darknet\data\OPGobj_data\IMG_1271.JPG	1	4-43000-pv340	B	boiler feedwater
C:\Repos\darknet\data\OPGobj_data\IMG_1272.JPG	2		B	
		4-43210-p3	C	main boiler feed pump
C:\Repos\darknet\data\OPGobj_data\IMG_1273.JPG	5	4-43220-v319	B	main boiler feed pump lubricating oil
		4-43220-v307	B	main boiler feed pump lubricating oil
		4-43220-v309	B	main boiler feed pump lubricating oil
		4-64322-pi306	B	main boiler feed pump lubricating oil
		4-43220-nv305	B	main boiler feed pump lubricating oil
C:\Repos\darknet\data\OPGobj_data\IMG_1274.JPG	2	4-72100-v623	B	low pressure service water (LPSW)
		4-72100-pv627	B	low pressure service water (LPSW)
C:\Repos\darknet\data\OPGobj_data\IMG_1275.JPG	1	4-5324-11cb4	A	boiler feed pump 4-4341pm3
C:\Repos\darknet\data\OPGobj_data\IMG_1276.JPG	3			
		4-5824-11cb6	A	condensate extraction pump 4- 4430-pm3

		4-5324-11cb5	A	circulating water pump 4-7112pm3
C:\Repos\darknet\data\OPGobj_data\IMG_1277.JPG	8			
		4-64323-ps2-v1	B	boiler feed pump gland seal
		4-64323-ps2	B	boiler feed pump gland seal
		4323-ps2-v1	B	boiler feed pump gland seal
		4-6-PL113-01	A	
		4-64323-pi1-v1003	B	boiler feed pump gland seal
		4-64323-pi1	B	boiler feed pump gland seal
C:\Repos\darknet\data\OPGobj_data\IMG_1278.JPG	2			
		4-64323-pl1467-11	A	
C:\Repos\darknet\data\OPGobj_data\IMG_1279.JPG	5			
		4-40-v50	B	boiler feed pump gland seal system
		4-43230-pv51	B	boiler feed pump gland seal system
		4-64323-sv2	B	boiler feed pump gland seal
C:\Repos\darknet\data\OPGobj_data\IMG_1280.JPG	1	0-75110-CPXX	A	Service air
C:\Repos\darknet\data\OPGobj_data\IMG_1281.JPG	4			
		3-67210-pi419	B	low pressure service water (LPSW)
		3-72100-v4613	B	low pressure service water (LPSW)
		3-67210-pi419-v1003	B	low pressure service water (LPSW)
C:\Repos\darknet\data\OPGobj_data\IMG_1282.JPG				
		0-75110-cp12	A	service air

		3-67210-pi419	B	low pressure service water (LPSW)
		3-67210-pi419-v1003	B	low pressure service water (LPSW)
		3-7	B	
		3-72100-v4613	B	low pressure service water (LPSW)
C:\Repos\darknet\data\OPGobj_data\IMG_1283.JPG	4	3-XX-7-PL1296	A	
		X-ZT201	B	generator control
		3-63617-zt202	B	steam generator level control
		3-63617-zt403	B	steam generator level control
C:\Repos\darknet\data\OPGobj_data\IMG_1284.JPG	2	3-67811-zs1	B	site fire protection water supply
		3-78110-v480	B	fire protection water supply distribn
C:\Repos\darknet\data\OPGobj_data\IMG_1285.JPG	1	3-67321-pl3519-01	A	
C:\Repos\darknet\data\OPGobj_data\IMG_1286.JPG	1	3-75120-rc7	C	inst. Air receiver
C:\Repos\darknet\data\OPGobj_data\IMG_1287.JPG	1	3-73220-hvu7	C	boiler rm hvu
C:\Repos\darknet\data\OPGobj_data\IMG_1288.JPG	1	3-42100-G	C	generator
C:\Repos\darknet\data\OPGobj_data\IMG_1289.JPG	1	4-75110	B	service air system
C:\Repos\darknet\data\OPGobj_data\IMG_1290.JPG				
C:\Repos\darknet\data\OPGobj_data\IMG_1291.JPG	1	4-72200tk1	C	rcw head tank
C:\Repos\darknet\data\OPGobj_data\IMG_1292.JPG	2	3-67811-nts1	A	dual feeds from
		3-67811-ds1	A	
C:\Repos\darknet\data\OPGobj_data\IMG_1387.JPG	2		B	
		0-75110-rv9201	B	service air

C:\Repos\darknet\data\OPGobj_data\IMG_1389.JPG	1	0-75110-rv9201	B	service air
C:\Repos\darknet\data\OPGobj_data\IMG_1390.JPG	5	0-75110-rv9004	B	service air
		0-75110-rv	B	
		0-67511-ds1	A	
		0-67511-pl5705	A	
		0-5334-mcc502-l3	A	fed from
C:\Repos\darknet\data\OPGobj_data\IMG_1391.JPG	3	0-5334-mcc502-l3	A	fed from
		0-67511-pl5705	A	
		0-67511-ds1	A	
C:\Repos\darknet\data\OPGobj_data\IMG_1393.JPG	3	0-75110-rd9201	B	service air
		0-75110-rv9204	B	service air
			B	
C:\Repos\darknet\data\OPGobj_data\IMG_1396.JPG	2	0-75110-rv9204	B	service air
		0-75110-rv9203	B	service air
C:\Repos\darknet\data\OPGobj_data\IMG_1408.JPG	1	0-76660-rv1	B	rv test bench
C:\Repos\darknet\data\OPGobj_data\IMG_1416.JPG	1	4-44030-v53	B	condensate system
C:\Repos\darknet\data\OPGobj_data\IMG_1418.JPG	1	4-44030-v52	B	condensate system
C:\Repos\darknet\data\OPGobj_data\IMG_1419.JPG	1	4-44030-or4	B	main condensate circuit
C:\Repos\darknet\data\OPGobj_data\IMG_1420.JPG	2			
		0-76650-pv1	B	rv test bench
C:\Repos\darknet\data\OPGobj_data\IMG_1423.JPG	1	3-44030-v53	B	main condensate circuit
C:\Repos\darknet\data\OPGobj_data\IMG_1424.JPG	1	3-44030-v53	B	main condensate circuit
C:\Repos\darknet\data\OPGobj_data\IMG_1429.JPG	1		B	
C:\Repos\darknet\data\OPGobj_data\IMG_1431.JPG	1		B	main condensate circuit
C:\Repos\darknet\data\OPGobj_data\IMG_1440.JPG	1	1-44030-v53	B	main condensate circuit
C:\Repos\darknet\data\OPGobj_data\IMG_1447.JPG	1	1-44030-v51	B	main condensate circuit
C:\Repos\darknet\data\OPGobj_data\IMG_1448.JPG	1	1-44030-v52	B	main condensate circuit
C:\Repos\darknet\data\OPGobj_data\IMG_1449.JPG	1	1-44030-v52	B	main condensate circuit

C:\Repos\darknet\data\OPGobj_data\IMG_1450.JPG	1	1-44030-v52	B	main condensate circuit
C:\Repos\darknet\data\OPGobj_data\IMG_1452.JPG	2			
		0-73260-DP9001	B	csa contam exhaust
C:\Repos\darknet\data\OPGobj_data\IMG_1476.JPG	2			
		0-73260-dp9200	B	csa venilation
C:\Repos\darknet\data\OPGobj_data\IMG_1478.JPG	4			
		0-73260-dp9201	B	csa contam exhaust
		0-73260-dp9200	B	csa contam exhaust
		0-73260-dp9001	B	csa contam exhaust
			B	csa contam exhaust
C:\Repos\darknet\data\OPGobj_data\IMG_1479.JPG	5			
		0-73260-dp9200	B	csa contam exhaust
		0-73260-dp9201	B	csa contam exhaust
		0-73260-dp9200	B	csa contam exhaust
		0-73260-dp9001	B	csa contam exhaust
C:\Repos\darknet\data\OPGobj_data\IMG_1482.JPG	1	0-73260-f9444	B	csa contam exhaust
C:\Repos\darknet\data\OPGobj_data\IMG_1483.JPG	1	0-56233-rp4277	A	from 0-56133-lp4027-cb13
C:\Repos\darknet\data\OPGobj_data\IMG_1502.JPG	2			
		0-67580-pi2	B	lab vacuum
C:\Repos\darknet\data\OPGobj_data\IMG_1503.JPG	3			
		0-67580-pi2	B	lab vacuum
		0-67580-prv20	B	lab vacuum
		0-67580-pi1	B	lab vacuum
C:\Repos\darknet\data\OPGobj_data\IMG_1504.JPG	1	0-75800-rv10	B	lab vacuum
C:\Repos\darknet\data\OPGobj_data\IMG_1508.JPG	3			
		0-67580-pi2	B	lab vacuum
		0-67580-prv20	B	lab vacuum
		0-67580-pi1	B	lab vacuum
C:\Repos\darknet\data\OPGobj_data\IMG_2210.JPG	2			
		1-44130-v6	B	condenser tube cleaning system
C:\Repos\darknet\data\OPGobj_data\IMG_2215.JPG	1	1-44130-v2	B	condenser tube cleaning system

C:\Repos\darknet\data\OPGobj_data\IMG_2216.JPG	1	1-44130-v6	B	condenser tube cleaning system
C:\Repos\darknet\data\OPGobj_data\IMG_2337.JPG	2	3-36140-rv2	C	bc1 isrv
		3	B	Ste
C:\Repos\darknet\data\OPGobj_data\IMG_2338.JPG	5		B	
		3-6036	B	steam
		3-63614-pi151	B	main steam supply
		3-3614	B	
		36140-rv2	C	bo1 isrv
C:\Repos\darknet\data\OPGobj_data\IMG_2339.JPG	4			
		3-6361	B	steam generator steam relief
		3-36140-nv9005	B	main steam supply
		3-63614-ps15l	B	main steam supply
C:\Repos\darknet\data\OPGobj_data\IMG_2340.JPG	3	3-36140-v9005	B	main steam supply
		15l	B	
		3-36	B	main steam supply
C:\Repos\darknet\data\OPGobj_data\IMG_2341.JPG	1	3-63614-zs15l	B	steam generator steam relief
C:\Repos\darknet\data\OPGobj_data\IMG_2460.JPG	1	0-72800-mv30	B	emergency service water
C:\Repos\darknet\data\OPGobj_data\IMG_2467.JPG	1	0-72800-mv35	B	emergency service water
C:\Repos\darknet\data\OPGobj_data\IMG_2720.JPG	1	0-63936-pl1978-21	A	
C:\Repos\darknet\data\OPGobj_data\IMG_2721.JPG	1	0-63936-AN2801	B	trf
C:\Repos\darknet\data\OPGobj_data\IMG_2722.JPG	2	0-63936-AN2801	B	trf
		0-63936-pl1978-21	A	
C:\Repos\darknet\data\OPGobj_data\IMG_2727.JPG	1	0-39450-v2423	B	trf gas sampling system
C:\Repos\darknet\data\OPGobj_data\IMG_2729.JPG	3	0-63336	B	
		0-39450-v2423	B	trf gas sampling system
		v2422	B	
C:\Repos\darknet\data\OPGobj_data\IMG_2740.JPG	4	0-63952-pi2	B	air clean-up

		0-63952-v1002	B	air clean-up
		0-63952-prv208	B	air clean-up
		0-39520-v214	B	trf cryo ref air clean up
C:\Repos\darknet\data\OPGobj_data\IMG_2741.JPG	2	0-63952-prv208	B	air clean-up
		0-6395	B	air clean-up
C:\Repos\darknet\data\OPGobj_data\Reheater.PNG	2	34-pau-102	C	reheater sys
			C	

## Appendix D

### Output from Auto-linking Algorithm containing GeoTag Information

	name	category	label	link	type	posx	posy	posz	visiByids	visiBynames	originalids
0	7111scl		7111scl	7111scl	hotlink	417	2743	1828	visiByids	Scans	originalids
1	7111scl		7111scl	7111scl	hotlink	417	2743	1828	visiByids	Scans	originalids
2	272200rv		272200rv	272200rv	hotlink	1097	1431	1291	visiByids	Scans	originalids
3	2722007134		2722007134	2722007134	hotlink	2139	893	1489	visiByids	Scans	originalids
4	2122007132		2122007132	2122007132	hotlink	1442	2876	747	visiByids	Scans	originalids
5	2122007132		2122007132	2122007132	hotlink	1442	2876	747	visiByids	Scans	originalids
6	2122007132		2122007132	2122007132	hotlink	1442	2876	747	visiByids	Scans	originalids
7	2722001137		2722001137	2722001137	hotlink	3299	1703	701	visiByids	Scans	originalids
8	2722001137		2722001137	2722001137	hotlink	3299	1703	701	visiByids	Scans	originalids
9	2161220tw		2161220tw	2161220tw	hotlink	494	1310	1695	visiByids	Scans	originalids
10	222ow		222ow	222ow	hotlink	1322	2184	770	visiByids	Scans	originalids
11	172200rv1336		172200rv1336	172200rv1336	hotlink	1690	1444	788	visiByids	Scans	originalids
12	172100v7011		172100v7011	172100v7011	hotlink	1570	1517	1428	visiByids	Scans	originalids
13	161517		161517	161517	hotlink	826	1492	1638	visiByids	Scans	originalids
14	167220p15111		167220p15111	167220p15111	hotlink	1352	672	1804	visiByids	Scans	originalids
15	167220p151v1		167220p151v1	167220p151v1	hotlink	1520	1206	1111	visiByids	Scans	originalids
16	167220p5111		167220p5111	167220p5111	hotlink	1641	2076	1228	visiByids	Scans	originalids
17	167220p1511		167220p1511	167220p1511	hotlink	2695	2095	775	visiByids	Scans	originalids
18	167220p1511		167220p1511	167220p1511	hotlink	2695	2095	775	visiByids	Scans	originalids
19	167220p1511		167220p1511	167220p1511	hotlink	2695	2095	775	visiByids	Scans	originalids
20	167220p1511		167220p1511	167220p1511	hotlink	2695	2095	775	visiByids	Scans	originalids
21	p15212v1003		p15212v1003	p15212v1003	hotlink	1087	1021	945	visiByids	Scans	originalids
22	p15212v1003		p15212v1003	p15212v1003	hotlink	1087	1021	945	visiByids	Scans	originalids



23	220pl1034		220pl1034	220pl1034	hotlink	1386	551	1520	visiByids	Scans	originalids
24	16120p15271		16120p15271	16120p15271	hotlink	2462	582	767	visiByids	Scans	originalids
25	1167220		1167220	1167220	hotlink	3137	2668	842	visiByids	Scans	originalids
26	161220p15211		161220p15211	161220p15211	hotlink	1483	9	666	visiByids	Scans	originalids
27	p152h2v0053		p152h2v0053	p152h2v0053	hotlink	1681	1363	872	visiByids	Scans	originalids
28	p152h2v0053		p152h2v0053	p152h2v0053	hotlink	1681	1363	872	visiByids	Scans	originalids
29	p152h1v1		p152h1v1	p152h1v1	hotlink	3016	2722	1312	visiByids	Scans	originalids
30	167220p15211		167220p15211	167220p15211	hotlink	3474	1026	1410	visiByids	Scans	originalids
31	167220		167220	167220	hotlink	2584	869	1426	visiByids	Scans	originalids
32	167220		167220	167220	hotlink	2552	1514	527	visiByids	Scans	originalids
33	167220		167220	167220	hotlink	2552	1514	527	visiByids	Scans	originalids
34	172200v73		172200v73	172200v73	hotlink	829	2531	688	visiByids	Scans	originalids
35	167220s711		167220s711	167220s711	hotlink	1651	1145	594	visiByids	Scans	originalids
36	172200vz		172200vz	172200vz	hotlink	3365	1811	413	visiByids	Scans	originalids
37	172200v13		172200v13	172200v13	hotlink	1124	2913	1084	visiByids	Scans	originalids
38	167220s731		167220s731	167220s731	hotlink	2923	225	1180	visiByids	Scans	originalids
39	112200p166		112200p166	112200p166	hotlink	1815	517	1326	visiByids	Scans	originalids
40	1453331n1515		1453331n1515	1453331n1515	hotlink	1743	1415	2065	visiByids	Scans	originalids
41	600		600	600	hotlink	2142	2515	1687	visiByids	Scans	originalids
42	600		600	600	hotlink	2142	2515	1687	visiByids	Scans	originalids
43	600		600	600	hotlink	2142	2515	1687	visiByids	Scans	originalids
44	600		600	600	hotlink	2142	2515	1687	visiByids	Scans	originalids
45	600		600	600	hotlink	983	1843	2704	visiByids	Scans	originalids
46	600		600	600	hotlink	983	1843	2704	visiByids	Scans	originalids
47	153331n1675		153331n1675	153331n1675	hotlink	2093	1177	1890	visiByids	Scans	originalids
48	600		600	600	hotlink	2158	2042	1956	visiByids	Scans	originalids
49	f1220otkz		f1220otkz	f1220otkz	hotlink	2331	1837	1025	visiByids	Scans	originalids
50	1723001196		1723001196	1723001196	hotlink	1110	1765	850	visiByids	Scans	originalids

51	112300pv46		112300pv46	112300pv46	hotlink	2297	1731	774	visiByids	Scans	originalids
52	172zovon		172zovon	172zovon	hotlink	1861	664	818	visiByids	Scans	originalids
53	172zovon		172zovon	172zovon	hotlink	1861	664	818	visiByids	Scans	originalids
54	172zovon		172zovon	172zovon	hotlink	1861	664	818	visiByids	Scans	originalids
55	172zovon		172zovon	172zovon	hotlink	1861	664	818	visiByids	Scans	originalids
56	1b1220pl644aoi		1b1220pl644aoi	1b1220pl644aoi	hotlink	1261	1759	1413	visiByids	Scans	originalids
57	1755o		1755o	1755o	hotlink	2239	1939	1537	visiByids	Scans	originalids
58	16723		16723	16723	hotlink	619	1402	626	visiByids	Scans	originalids
59	pi34vio03		pi34vio03	pi34vio03	hotlink	1498	2718	445	visiByids	Scans	originalids
60	pi34vio03		pi34vio03	pi34vio03	hotlink	1498	2718	445	visiByids	Scans	originalids
61	1167323		1167323	1167323	hotlink	2482	2707	520	visiByids	Scans	originalids
62	167328		167328	167328	hotlink	3771	2711	386	visiByids	Scans	originalids
63	167328		167328	167328	hotlink	3771	2711	386	visiByids	Scans	originalids
64	167517v42		167517v42	167517v42	hotlink	1926	83	1459	visiByids	Scans	originalids
65	16751		16751	16751	hotlink	2433	2472	1921	visiByids	Scans	originalids
66	1666li06a		1666li06a	1666li06a	hotlink	1848	1574	2623	visiByids	Scans	originalids
67	1o71zrva2n		1o71zrva2n	1o71zrva2n	hotlink	687	2589	1071	visiByids	Scans	originalids
68	1o71zrva2n		1o71zrva2n	1o71zrva2n	hotlink	687	2589	1071	visiByids	Scans	originalids
69	3641		3641	3641	hotlink	2170	1044	1350	visiByids	Scans	originalids
70	36xxjbi6t71		36xxjbi6t71	36xxjbi6t71	hotlink	2166	1576	945	visiByids	Scans	originalids
71	36120sisn		36120sisn	36120sisn	hotlink	1874	2050	378	visiByids	Scans	originalids
72	36120sisn		36120sisn	36120sisn	hotlink	1874	2050	378	visiByids	Scans	originalids
73	361210fs15775		361210fs15775	361210fs15775	hotlink	2501	3007	617	visiByids	Scans	originalids
74	361210fs15775		361210fs15775	361210fs15775	hotlink	2501	3007	617	visiByids	Scans	originalids
75	361210fs15775		361210fs15775	361210fs15775	hotlink	2501	3007	617	visiByids	Scans	originalids
76	3486otk		3486otk	3486otk	hotlink	2906	1423	877	visiByids	Scans	originalids
77	4533421cbio		4533421cbio	4533421cbio	hotlink	1775	2564	1087	visiByids	Scans	originalids
78	4533421cbio		4533421cbio	4533421cbio	hotlink	1775	2564	1087	visiByids	Scans	originalids

79	4a30oopv340		4a30oopv340	4a30oopv340	hotlink	2686	1681	641	visiByids	Scans	originalids
80	4a30oopv340		4a30oopv340	4a30oopv340	hotlink	2686	1681	641	visiByids	Scans	originalids
81	1507		1507	1507	hotlink	1424	3165	430	visiByids	Scans	originalids
82	aa3220v309		aa3220v309	aa3220v309	hotlink	1450	1856	390	visiByids	Scans	originalids
83	4fi122p1306		4fi122p1306	4fi122p1306	hotlink	1599	962	319	visiByids	Scans	originalids
84	4fi122p1306		4fi122p1306	4fi122p1306	hotlink	1599	962	319	visiByids	Scans	originalids
85	477100		477100	477100	hotlink	1741	329	959	visiByids	Scans	originalids
86	477100		477100	477100	hotlink	1741	329	959	visiByids	Scans	originalids
87	46324icba		46324icba	46324icba	hotlink	2135	1456	1629	visiByids	Scans	originalids
88	453241icb5		453241icb5	453241icb5	hotlink	3402	1194	1369	visiByids	Scans	originalids
89	4b4bzbpli6ta		4b4bzbpli6ta	4b4bzbpli6ta	hotlink	2213	1620	742	visiByids	Scans	originalids
90	443230p15		443230p15	443230p15	hotlink	3168	2087	410	visiByids	Scans	originalids
91	361210p14119		361210p14119	361210p14119	hotlink	2622	1608	517	visiByids	Scans	originalids
92	361210p14119		361210p14119	361210p14119	hotlink	2622	1608	517	visiByids	Scans	originalids
93	361210p14119		361210p14119	361210p14119	hotlink	2622	1608	517	visiByids	Scans	originalids
94	70l1296		70l1296	70l1296	hotlink	351	55	2789	visiByids	Scans	originalids
95	363617z1403		363617z1403	363617z1403	hotlink	131	1821	1069	visiByids	Scans	originalids
96	363617z7202		363617z7202	363617z7202	hotlink	2533	1587	668	visiByids	Scans	originalids
97	361321pl361900		361321pl361900	361321pl361900	hotlink	2280	1332	1231	visiByids	Scans	originalids
98	375120rci		375120rci	375120rci	hotlink	2405	1887	1086	visiByids	Scans	originalids
99	pl103521		pl103521	pl103521	hotlink	35	792	724	visiByids	Scans	originalids
100	461220ltan		461220ltan	461220ltan	hotlink	628	2590	482	visiByids	Scans	originalids
101	467220l		467220l	467220l	hotlink	1026	1832	528	visiByids	Scans	originalids
102	467220l		467220l	467220l	hotlink	1238	2563	419	visiByids	Scans	originalids
103	467220l		467220l	467220l	hotlink	1238	2563	419	visiByids	Scans	originalids
104	467220		467220	467220	hotlink	2088	2114	500	visiByids	Scans	originalids
105	467220		467220	467220	hotlink	2088	2114	500	visiByids	Scans	originalids
106	467220		467220	467220	hotlink	2749	2042	615	visiByids	Scans	originalids

107	467220		467220	467220	hotlink	2749	2042	615	visiByids	Scans	originalids
108	167220p155		167220p155	167220p155	hotlink	3772	1478	392	visiByids	Scans	originalids
109	31678111ds1		31678111ds1	31678111ds1	hotlink	2480	1086	403	visiByids	Scans	originalids
110	067511pl5705		067511pl5705	067511pl5705	hotlink	1705	1057	479	visiByids	Scans	originalids
111	067511ds1		067511ds1	067511ds1	hotlink	3133	963	408	visiByids	Scans	originalids
112	aao3ov52		aao3ov52	aao3ov52	hotlink	1484	2707	584	visiByids	Scans	originalids
113	1403ovs2		1403ovs2	1403ovs2	hotlink	286	3538	2601	visiByids	Scans	originalids
114	562331		562331	562331	hotlink	1288	727	980	visiByids	Scans	originalids
115	0510212017		0510212017	0510212017	hotlink	3953	3210	972	visiByids	Scans	originalids
116	0510212017		0510212017	0510212017	hotlink	2443	4541	1027	visiByids	Scans	originalids
117	281191280y		281191280y	281191280y	hotlink	1133	1519	1469	visiByids	Scans	originalids
118	281199270y		281199270y	281199270y	hotlink	1146	1457	1408	visiByids	Scans	originalids
119	1991280y		1991280y	1991280y	hotlink	1650	1601	2552	visiByids	Scans	originalids
120	364orv		364orv	364orv	hotlink	2342	1887	469	visiByids	Scans	originalids
121	36361		36361	36361	hotlink	1392	50	873	visiByids	Scans	originalids
122	0612612017		0612612017	0612612017	hotlink	3817	3214	1215	visiByids	Scans	originalids
123	ool2012017z		ool2012017z	ool2012017z	hotlink	4033	3217	814	visiByids	Scans	originalids
124	c6ibban01		c6ibban01	c6ibban01	hotlink	1850	1533	1314	visiByids	Scans	originalids
125	06666pli162		06666pli162	06666pli162	hotlink	1784	434	1582	visiByids	Scans	originalids
126	063952v1002		063952v1002	063952v1002	hotlink	3237	1793	561	visiByids	Scans	originalids
127	0375201214		0375201214	0375201214	hotlink	3560	2763	749	visiByids	Scans	originalids
128	0375201214		0375201214	0375201214	hotlink	3560	2763	749	visiByids	Scans	originalids

## Appendix E

### Python Script for Correcting OCR Detected Text

```
# -*- coding: utf-8 -*-
"""
Created on Fri Feb 19 12:57:17 2021
This code is intended as a text recommender
@author: c7edward
"""
import nltk
nltk.download('words')
import pandas as pd

text = open("database.txt")
cols_list = ["name"]
results_ocr = pd.read_csv("output.csv")
entry_ocr = results_ocr["name"].tolist()
print(entry_ocr)

list1 = text.readlines()
list2 = [x.replace('\n', '') for x in list1]

print(results_ocr)
print(list2)

spellings_series = pd.Series(list2)

def getcorrections(ocr_entries, database):
    correct_list = []
    for entry in ocr_entries:
        distance = 99
        correction = ""
        for data in database:
```

```
            ed = nltk.edit_distance(data, entry)
            if ed < distance:
                distance = ed
                correction = data
            correct_list.append(correction)
    return correct_list

print(entry_ocr)
print(getcorrections(entry_ocr, list2))

results_ocr['corrections'] =
getcorrections(entry_ocr, list2)
results_ocr.to_csv('output_corrected.csv', index=False
)
```

The output from the text correction is shown below. Compared to the “name” column the corrections are more accurate. The other columns of the output.csv were removed to simplify this table format.

image	name	corrections	Actual	Match?
IMG_1127.JPG	7111scl	7111-sc2	7111-sc2	Match
IMG_1127.JPG	7111scl	7111-sc2	7111-sc2	Match
IMG_1205.JPG	272200rv	2-72200-rv138	2-72200-rv138	Match
IMG_1206.JPG	2722007134	2-72200-v134	2-72200-v134	Match
IMG_1207.JPG	2122007132	2-72200-v132	2-72200-v132	Match
IMG_1207.JPG	2122007132	2-72200-v132	2-72200-v132	Match
IMG_1207.JPG	2122007132	2-72200-v132	2-72200-v132	Match
IMG_1207.JPG	2722001137	2-72200-v137	2-72200-v137	Match
IMG_1207.JPG	2722001137	2-72200-v137	2-72200-v137	Match
IMG_1208.JPG	2161220tw	2-67220-tw151	2-67220-tw152	No Match
IMG_1208.JPG	222ow	2-722	2-67220-tw151	No Match
IMG_1209.JPG	172200rv1336	2-72200-rv133	1-72200-rv138	No Match
IMG_1212.JPG	172100v7011	1-72100-v3013	1-72100-v3013	Match
IMG_1214.JPG	161517	15l	1-67517-prv31-4	No Match
IMG_1215.JPG	167220p15111	1-67220-pi51	1-67220-pi51	Match
IMG_1215.JPG	167220p151v1	1-67220-pi51-v1	1-67220-pi51-v1	Match
IMG_1215.JPG	167220p5111	1-67220-pi51	1-67220-pi51-v1	No Match
IMG_1215.JPG	167220p1511	1-67220-pi51	1-67220-pi53	No Match
IMG_1215.JPG	167220p1511	1-67220-pi51	1-67220-pi53	No Match
IMG_1215.JPG	167220p1511	1-67220-pi51	1-67220-pi53-v1	No Match
IMG_1215.JPG	167220p1511	1-67220-pi51	1-67220-pi51	Match
IMG_1216.JPG	p15212v1003	s152-v1	1-67220-pi52#1-v1	No Match
IMG_1216.JPG	p15212v1003	s152-v1	1-67220-pi52#1-v1	No Match
IMG_1216.JPG	220pl1034	2-72200-v134	1-72200-v9047	No Match
IMG_1216.JPG	16120p15271	1-67220-pi52#1	1-67220-pi52#1	Match
IMG_1216.JPG	1167220	1-67323	1-67220-pi52#2-v1003	No Match
IMG_1217.JPG	161220p15211	1-67220-pi52#1	1-67220-pi52#1	Match
IMG_1217.JPG	p152h2v0053	s152-v1	1-67220-pi52#1-v1	No Match
IMG_1217.JPG	p152h2v0053	s152-v1	1-67220-pi52#2-v1003	No Match
IMG_1217.JPG	p152h1v1	s152-v1	0-pl1034-01	No Match
IMG_1217.JPG	167220p15211	1-67220-pi52#1	1-67220-pi52#1	Match
IMG_1222.JPG	167220	2-722	1-67220-te12#2	No Match
IMG_1223.JPG	167220	2-722	1-67220-te12#2	No Match
IMG_1223.JPG	167220	2-722	1-67220-te12#2	No Match
IMG_1226.JPG	172200v73	1-72200-v73	1-72200-v73	Match
IMG_1226.JPG	167220s711	1-72200-p1	1-67220-sv31	No Match

IMG_1226.JPG	172200vz	1-72200-p1	1-72200-v72	No Match
IMG_1227.JPG	172200v13	1-72200-v73	1-72200-v73	Match
IMG_1227.JPG	167220s731	1-67220-sv31	1-67220-sv31	Match
IMG_1228.JPG	112200p166	1-72200-pv66	1-72200-pv66	Match
IMG_1229.JPG	1453331n1515	1-5333-cn-15-25	1-5333-cn-15-25	Match
IMG_1229.JPG	600	RV1		No Match
IMG_1229.JPG	600	RV1		No Match
IMG_1229.JPG	600	RV1		No Match
IMG_1229.JPG	600	RV1		No Match
IMG_1230.JPG	600	RV1		No Match
IMG_1230.JPG	600	RV1		No Match
IMG_1230.JPG	153331n1675	1-5333-cn-16-25	1-5333-cn-16-25	Match
IMG_1230.JPG	600	RV1		No Match
IMG_1232.JPG	f1220otkz	4-72200tk1	1-72200-tk2	No Match
IMG_1233.JPG	1723001196	1-72300-v195	1-72300-v195	Match
IMG_1235.JPG	112300pv46	1-72300-pv46	1-72300-pv46	Match
IMG_1237.JPG	172zovon	s152-v1	1-67220-v9026	No Match
IMG_1237.JPG	172zovon	s152-v1	1-67220-v9019	No Match
IMG_1237.JPG	172zovon	s152-v1	1-67220-v9024	No Match
IMG_1237.JPG	172zovon	s152-v1		No Match
IMG_1239.JPG	1b1220pl644aoi	1-72100-v447	1-67220-pl3444-01	No Match
IMG_1240.JPG	1755o	15l	1-73230-tk1	No Match
IMG_1241.JPG	16723	1-67323	1-67323-pi35-v1007	No Match
IMG_1241.JPG	pi34vio03	13210	1-67323-pi34-v1003	No Match
IMG_1241.JPG	pi34vio03	13210	1-67323-pi34	No Match
IMG_1241.JPG	1167323	1-67323	1-67323-pi35-v1003	No Match
IMG_1241.JPG	167328	1-67323	1-67323-pi6-v1003	No Match
IMG_1241.JPG	167328	1-67323	1-67323-pi6-v1003	No Match
IMG_1246.JPG	167517v42	1-7110-sc2	1-67517-rv14-12	No Match
IMG_1246.JPG	16751	15l	1-67517-tk14-11	No Match
IMG_1248.JPG	1666li06a	1-67323	1-63862-pl10084-11	No Match
IMG_1249.JPG	1o71zrva2n	1-7110-sc2	1-67517-prv14-24	No Match
IMG_1249.JPG	1o71zrva2n	1-7110-sc2	1-67517-prv14-23	No Match
IMG_1257.JPG	3641	RV1	3-64184-lcv49#1-v900	No Match
IMG_1260.JPG	36xxjbi6t71	3-48500-tk1	3-67322-jb1670-21	No Match
IMG_1262.JPG	36120sisn	36140-rv2	s152-v1	No Match
IMG_1262.JPG	36120sisn	36140-rv2	3-67210-fs153-v5	No Match
IMG_1262.JPG	361210fs15775	3-67210-fs153-v1	3-67210-fs151-v1	No Match
IMG_1262.JPG	361210fs15775	3-67210-fs153-v1	3-67210-fs152-v1	No Match
IMG_1262.JPG	361210fs15775	3-67210-fs153-v1	3-67210-fs153-v1	Match
IMG_1266.JPG	34860otk	3-48500-tk1	3-48500-tk1	Match

IMG_1267.JPG	4533421cbio	4-5534-27bio	4-5334-27bio	No Match
IMG_1267.JPG	4533421cbio	4-5534-27bio	4-3333-htr5	No Match
IMG_1271.JPG	4a30oopv340	4-43000-pv340	4-43000-pv340	Match
IMG_1271.JPG	4a30oopv340	4-43000-pv340	4-43000-pv340	Match
IMG_1273.JPG	1507	15l	4-43-07	No Match
IMG_1273.JPG	aa3220v309	4-43220-v309	4-43220-v309	Match
IMG_1273.JPG	4fi122p1306	4-64322-pi306	4-64322-pi306	Match
IMG_1273.JPG	4fi122p1306	4-64322-pi306	4-43220-nv306	No Match
IMG_1274.JPG	477100	4-75110	4-72100-v623	No Match
IMG_1274.JPG	477100	4-75110	4-72100-pv627	No Match
IMG_1275.JPG	46324icba	4-5324-11cb4	4-5324-11cb4	Match
IMG_1276.JPG	453241icb5	4-5324-11cb5	4-5324-11cb5	Match
IMG_1278.JPG	4b4bzbpli6ta	4-47100-tk1	4-64323-pl1467-11	No Match
IMG_1279.JPG	443230p15	4-43230-pv51	4-43230-pv51	Match
IMG_1282.JPG	361210p14119	3-67210-pi419	3-67210-pi419	Match
IMG_1282.JPG	361210p14119	3-67210-pi419	0-75110-cp12	No Match
IMG_1282.JPG	361210p14119	3-67210-pi419	3-67210-pi419	Match
IMG_1283.JPG	70l1296	0-6395	3-67-pl1296	No Match
IMG_1283.JPG	363617z1403	3-63617-zt403	3-63617-zt403	Match
IMG_1283.JPG	363617z7202	3-63617-zt202	3-63617-zt202	Match
IMG_1285.JPG	361321pl361900	3-67321-pl3519-01	3-67321-pl3519-01	Match
IMG_1286.JPG	375120rci	3-75120-rc7	3-75120-rc7	Match
IMG_1290.JPG	pl103521	40-RV1	4-67220-lt34-v1	No Match
IMG_1290.JPG	461220ltan	4-72200tk1	4-67220-lt34-v1	No Match
IMG_1290.JPG	467220l	2-722	4-67220-lt34-v1	No Match
IMG_1290.JPG	467220l	2-722	4-67220-ps35#2-v1003	No Match
IMG_1290.JPG	467220l	2-722	4-67220-lt34	No Match
IMG_1290.JPG	467220	2-722	4-67220-ps35#2	No Match
IMG_1290.JPG	467220	2-722	4-67220-ps35#1-v1003	No Match
IMG_1290.JPG	467220	2-722	4-67220-pi35	No Match
IMG_1290.JPG	467220	2-722	pl1035-21	No Match
IMG_1290.JPG	167220p155	1-67220-pi51	3-67811-ds1	No Match
IMG_1292.JPG	31678111ds1	3-67811-ds1	3-67811-ds1	Match
IMG_1391.JPG	067511pl5705	0-67511-pl5705	0-67511-pl5705	Match
IMG_1391.JPG	067511ds1	0-67511-ds1	0-67511-ds1	Match
IMG_1448.JPG	aao3ov52	4-40-v50	1-44030-v52	No Match
IMG_1449.JPG	1403ovs2	1-44030-v52	1-44030-v52	Match
IMG_1483.JPG	562331	22341	0-56233-rp4277	No Match
IMG_2210.JPG	510212017	0-67516-v2317	1-44130-v6	No Match



IMG_2215.JPG	510212017	0-67516-v2317	1-44130-v2	No Match
IMG_2337.JPG	281191280y	7111-sc2	3-36140-rv2	No Match
IMG_2337.JPG	281199270y	7111-sc2	3-36140-rv2	No Match
IMG_2337.JPG	1991280y	X-ZT201	3-36140-rv2	No Match
IMG_2338.JPG	364orv	3-64184	3-6140-rv2	No Match
IMG_2339.JPG	36361	22341	3-6361-	No Match
IMG_2460.JPG	612612017	0-67516-v2317	0-72800-mv30	No Match
IMG_2467.JPG	ool2012017z	X-ZT201	0-72800-14v35	No Match
IMG_2721.JPG	c6ibban01	X-ZT201	0-63936-AN2801	No Match
IMG_2722.JPG	06666pli162	0-76660-rv1	0-63936-pl1978-21	No Match
IMG_2740.JPG	063952v1002	0-63952-v1002	0-63952-v1002	Match
IMG_2740.JPG	375201214	0-39520-v214	0-39520-v214	Match
IMG_2740.JPG	375201214	0-39520-v214	0-63952-pi2	No Match