

Improving automation in computer vision based indoor construction progress monitoring: A deep learning approach

Biyanka Jayangani Ekanayake

Thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

under the supervision of

Associate Professor Johnny Wong

Associate Professor Alireza Fini

Associate Professor Peter Smith

University of Technology Sydney

Faculty of Design Architecture and Building

December 2022

Certificate of original authorship

I, Biyanka Jayangani Ekanayake, declare that this thesis is submitted in fulfilment of the requirements for the award of the degree of Doctor of Philosophy, in the School of Built Environment of the Faculty of Design, Architecture and Building at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Production Note:

Signature: Signature removed prior to publication.

Date: 29 December 2022

Note on the thesis format

This thesis was prepared to fulfil the criteria for Doctor of Philosophy and is in the format of a conventional thesis. It adheres to the guidelines set forth in “Graduate Research Candidature Management, Thesis Preparation and Submission Procedures 2022 - University of Technology Sydney”. This thesis used the American Psychological Association (APA) 7th edition referencing style. In Chapter 2: Literature Review, the source of the figures created by the thesis author (Biyanka Jayangani Ekanayake) is cited as “The Author”.

Acknowledgement

First and foremost, I would like to express my gratitude to my amazing supervisors. My principal supervisor, Associate Professor Johnny Wong gave me freedom and valuable insights for innovation in my research with constant encouragement. I sincerely thank him for being there for me not only to provide constructive feedback for my write-ups but also to listen to my typical PhD despairs, always with a kind smile. I am grateful to my supervisor, Associate Professor Alireza Fini for giving valuable insights, which made me work harder and think more critically. He even accompanied me to a construction site to ensure that everything went smoothly for my data collection during the COVID pandemic. My supervisor, Associate Professor Peter Smith kindly guided me to see my research from different perspectives, especially when I was too invested in the algorithmic part. My heartfelt gratitude goes to him for inspiring me to resolve issues with a calm approach. I have been blessed to be guided by such kind, supportive and understanding supervisors.

I am grateful to the building contractor organisation and Benjamin Lunn for their great support during the data collection process. Without Ben's support, the access to the construction sites and setting up of cameras would have been impossible. I express my special thanks to UTS 5D BIM Lab for employing me as a research assistant and also providing additional financial support for the research equipment I needed. My gratitude is extended to School of Built Environment and Faculty of Design, Architecture and Building of UTS for providing the research space and great facilities to complete my thesis. I would also like to thank the DAB HDR Director Associate Professor Toby Slade, former HDR Director Associate Professor Michael Er, DAB HDR Support team, Georgina Donovan, and Vanessa Setiadi and the Graduate Research School for all the support during my candidature.

My heartfelt thanks goes to many wonderful people I met in Sydney and UTS. Thank you, Adi for being the friend I needed when I first came to Sydney. My dearest friends, Mia and Tazy, thank you for making me part of your families and taking me on many trips in Sydney. Big thanks to Mona for her valuable advice. Since I started my PhD journey, I have been blessed to have incredible friends like Elham, Mukesh, Megan, Bingyang, Mani, Mirka, Javid, Enya, Caroline, Fiona, Samuel, Xiaohang, Juanwen and

rest of the DAB PhD Girls et al. group. I never felt alone in my PhD journey and in Sydney because of these fellow PhD friends.

My biggest gratitude goes to my parents for the unconditional love, support, wonderful upbringing, and opportunities they have given me. I could not have done any of this without their blessing. I am forever indebted to my beloved brother Thilina for taking care of everything and everyone while I was in Sydney for 3 years working on my PhD. With great love, I want to thank my boyfriend Ashan for his patience, understanding, care and support.

Table of contents

Certificate of original authorship	i
Note on the thesis format	ii
Acknowledgement.....	iii
Table of contents	v
List of tables.....	viii
List of figures	viii
Abbreviations	x
Peer reviewed publications from this research.....	xii
Abstract	xiii
1 Introduction.....	1
1.1 Background	1
1.2 Problem statement	4
1.3 Research questions	5
1.4 Research aim and objectives	5
1.5 Overview of the research methodology.....	6
1.6 Scope and limitations	6
1.7 Outline of the thesis.....	8
2 Literature review.....	10
2.1 Challenges of traditional construction progress monitoring	10
2.2 State-of-the-art progress monitoring approaches	12
2.2.1 Laser scanning based technologies	14
2.2.2 Radio wave based technologies	15
2.2.3 Geospatial based technologies	17
2.2.4 Computer vision based technologies.....	18
2.3 Algorithmic approaches to analyse the as-built status from image data	24
2.3.1 Conventional machine learning approaches.....	24
2.3.2 Deep learning approaches	27
2.3.3 Convolutional neural networks	29
2.4 Indoor construction progress monitoring	31
2.4.1 Advances in indoor image collection and visualisation.....	35
2.4.2 Automated activity state recognition of indoor works.....	37
2.4.3 Adoption of deep learning in indoor construction progress monitoring ...	39
2.5 Research gaps	42
2.5.1 Technical challenges for as-built indoor elements visual recognition	42

2.5.2	The deep learning model to overcome the challenges	48
2.5.3	Calculating work-in-progress using the deep learning model.....	50
2.6	Chapter summary	52
3	Research design and methods	54
3.1	Determining the methodological framework	54
3.1.1	Design science research	55
3.1.2	Research process mapped with the design science research framework ..	56
3.2	Data collection from the case projects	59
3.2.1	Case projects	59
3.2.2	Time-lapse cameras, related accessories, and camera settings	61
3.2.3	Developing the training images dataset	66
3.3	Research design for developing the deep learning approach	70
3.3.1	Mask R-CNN network architecture	70
3.3.2	Building the deep learning model: Transfer learning and fine-tuning	72
3.3.3	Training platform for the deep learning model: Colab virtual machine ...	74
3.3.4	Metrics to evaluate the performance of deep learning models	76
3.4	Chapter summary	77
4	Results and discussion: Part I	80
4.1	Empirical evidence from the case projects.....	80
4.1.1	Indoor objects related challenges	82
4.1.2	Lighting conditions related challenges.....	85
4.1.3	Camera positioning related challenges.....	88
4.2	Pre-processing techniques to reduce the impacts of the technical challenges..	92
4.2.1	Perspective transformation in image processing.....	93
4.2.2	Low illumination enhancement in image processing.....	94
4.3	Discussion	98
4.4	Chapter summary	101
5	Results and discussion: Part II	103
5.1	Automated visual recognition of indoor as-built elements.....	103
5.1.1	The workflow of modifying Mask R-CNN and training on Colab.....	103
5.1.2	Performance evaluation of the optimised Mask R-CNN model	111
5.2	Calculating the work-in-progress of indoor elements	117
5.3	Discussion	123
5.4	Chapter summary	126
6	Conclusions and recommendations	128

6.1	Achieving the research objectives and summary of the outcomes.....	128
6.1.1	Investigating the technical challenges and their impacts for indoor as-built elements visual recognition	129
6.1.2	Developing an optimised deep learning algorithm for automated visual recognition of indoor as-built elements	130
6.1.3	Automatically calculating the percentage work completion of indoor elements by post-processing the optimised deep learning algorithm	132
6.2	Theoretical contributions and practical implications	133
6.2.1	Contributions to the body of knowledge.....	133
6.2.2	Implications for the construction industry	134
6.3	Limitations of the current study and future research directions.....	136
	References	139
	Appendix 1 Research ethics application approval	156

List of tables

Table 2.1: Data acquisition technologies for progress monitoring	23
Table 2.2: Overview of computer vision based indoor construction monitoring	32
Table 2.3: Technical challenges for visual recognition of as-built indoor elements	46
Table 3.1: Information about the case projects	61
Table 3.2: Time-lapse camera settings.....	65
Table 4.1: Technical challenges for as-built indoor elements visual recognition.....	99
Table 5.1: Customising certain hyperparameters.....	110

List of figures

Figure 1.1: Mapping of research objectives, methods and corresponding chapters	8
Figure 2.1: Process of automated progress monitoring in construction projects	12
Figure 2.2: Trimble GX 3D laser scanner	14
Figure 2.3: 3D as-built reconstruction of a building using point cloud	15
Figure 2.4: RFID and barcode technologies	16
Figure 2.5: GPS data logger	18
Figure 2.6: A construction site a) 2D image; b) 3D photogrammetric model	20
Figure 2.7: 2D site images integrated with 4D BIM.....	21
Figure 2.8: Feature extraction: a) Original image; b) Subjected to Canny edge detection	25
Figure 2.9: Image segmentation: a) Original image; b) Binary image	26
Figure 2.10: Mechanisms: a) Traditional machine learning; b) Deep learning	27
Figure 2.11: Mechanism behind deep neural networks	29
Figure 2.12: Typical structure of a CNN and the functions of the hidden layers	30
Figure 2.13: Timeline of the computer vision based indoor progress monitoring studies	42
Figure 3.1: Process of design science research	57
Figure 3.2: Case projects: a) Project 1; b) Project 2; c) Project 3	60
Figure 3.3: Brinno TLC200 Pro time-lapse camera.....	62
Figure 3.4: Installing cameras: a) Using a tripod; b) Taping to scaffolding	63
Figure 3.5: Camera accessories: a) Gorillapod; b) Weather resistant housing	64
Figure 3.6: Indoor location with a) Natural lighting; b) Artificial lighting	64
Figure 3.7: a) Image captured from Project 1; b) Image sourced from the internet	67
Figure 3.8: Sample of Python code for image augmentation.....	68
Figure 3.9: a) Original image; b) Image flipped vertically and contrast enhanced.....	69
Figure 3.10: Annotated image.....	70
Figure 3.11: Mask R-CNN network architecture	71
Figure 3.12: Illustrating transfer learning	73
Figure 3.13: Working mechanism of Colab	75
Figure 3.14: Summary of the research process	78
Figure 4.1: Sample of Python code for Canny edge detection.....	81
Figure 4.2: Construction equipment creating occlusions.....	82
Figure 4.3: Construction personnel and material creating occlusions	83
Figure 4.4: Movement of the drywall sheets.....	84

Figure 4.5: a) Original image; b) Canny edge detector extracting insulation and drywall states.....	84
Figure 4.6: Low illumination in the indoor site	85
Figure 4.7: Backlight caused by natural light entering the site.....	86
Figure 4.8: Extracting the region of interest affected by a bright artificial lighting	87
Figure 4.9: a) Original image; b) Subjected to Canny edge detection	87
Figure 4.10: The slightly inclined region of interest captured by the camera.....	88
Figure 4.11: Field of view is affected by camera positioning.....	89
Figure 4.12: Camera location is far from the walls.....	89
Figure 4.13: Obstruction to the field of view of the camera	90
Figure 4.14: Camera being accidentally turned away	90
Figure 4.15: a) Original image; b) Original subjected to Canny edge detection	91
Figure 4.16: Region of interest subjected to Canny edge detection.....	92
Figure 4.17: Python code for perspective transformation.....	93
Figure 4.18: Perspective transformation: a) Original image; b) Corrected image.....	94
Figure 4.19: Python code for low light enhancement: Part I	96
Figure 4.20: Python code for low light enhancement: Part II.....	97
Figure 4.21: Low illumination enhancement: a) Original image; b) Enhanced image... ..	98
Figure 5.1: The technical algorithm	104
Figure 5.2: Mounting Google Drive to Colab.....	105
Figure 5.3: The specifications of the Colab virtual machine	106
Figure 5.4: Installing Detectron2	107
Figure 5.5: Importing some common libraries and PyTorch related utilities	107
Figure 5.6: Loading the dataset.....	108
Figure 5.7: Loading the backbone and fine-tuning the hyperparameters.....	109
Figure 5.8: Setting up the directory to save the trained weights.....	110
Figure 5.9: Training the model.....	110
Figure 5.10: Comparison of mAPs with different learning rates	112
Figure 5.11: Comparison of training loss with different learning rates	113
Figure 5.12: Comparison of validation loss with different learning rates.....	113
Figure 5.13: Training loss and validation loss of the model.....	114
Figure 5.14: Loss: a) Bounding box loss; b) Class loss; c) Mask loss; d) Total loss....	115
Figure 5.15: Inferring using the trained model	116
Figure 5.16: Examples of indoor as-built state recognition using Mask R-CNN	116
Figure 5.17: a) Image of an indoor wall; b) Segmented mask area calculations	118
Figure 5.18: Mounting Google Drive	119
Figure 5.19: Importing the libraries	120
Figure 5.20: Test images and output directories	120
Figure 5.21: Pixel masks area calculation.....	121
Figure 5.22: a) Work completion; b) As-planned wall area from CAD model	122
Figure 5.23: Indoor partition wall is fully installed	123
Figure 6.1: Contributions of the PhD study	136

Abbreviations

ALS	Airborne laser scanner
AP	Average precision
AVI	Audio Video Interleave
AU\$	Australian dollars
BIM	Building information model
CAD	Computer aided design
CDF	Cumulative distribution function
CLAHE	Contrast-limited adaptive histogram equalisation
CNN	Convolutional neural network
COCO	Common objects in context
CUDA	Compute unified device architecture
cuDNN	CUDA based deep neural networks
CV	Computer vision
FC	Fully connected
FCN	Fully convolutional net
FN	False negatives
FP	False positives
FPN	Feature pyramid network
FPS	Frames per second
GIS	Geographic information system
GNSS	Global navigation satellite system
GPS	Global positioning system
GPU	Graphic processing unit
HD	High definition
HDR	High dynamic range
HOG	Histogram of oriented gradients
IMU	Initial measurement unit
IoU	Intersection over union
JPG	Joint Photographic Experts Group
LBP	Local binary patterns
LiDAR	Light detection and ranging
mAP	Mean average precision
Mask R-CNN	Mask regional convolutional neural network
MLS	Mobile laser scanner
MVS	Multi view stereo
NSW	New South Wales
OpenCV	Open source computer vision
PDA	Personal digital assistant
PPHT	Progressive probabilistic Hough Transform
PSPNet	Pyramid scene parsing network
RANSAC	Random sample consensus
R-CNN	Region-based convolutional neural networks

ReLU	Rectified linear unit
ResNet	Residual network
RFID	Radio frequency identification
RGB-D	Red, green, and blue colour channels
RPN	Region proposal network
SFM	Structure from motion
SGD	Stochastic gradient descent
SLAM	Simultaneous localisation and mapping
SSD	Single shot multi-box detector
SVM	Support vector machine
TLS	Terrestrial laser scanner
TN	True negative
TP	True positive
TPU	Tensor processing unit
UAS	Unmanned aerial system
UAV	Unmanned aerial vehicle
UGV	Unmanned ground vehicle
US\$	United States dollars
UWB	Ultra-wide band
VO	Visual odometry
VSLAM	Visual simultaneous localisation and mapping
YOLO	You Only Look Once
YOLOCT++	You Only Look at CoefficientT++

Peer reviewed publications from this research

Ekanayake, B., Wong, J. K. W., Fini, A. A. F., and Smith, P. (2021), Computer vision based interior construction progress monitoring: A literature review and future research directions. *Automation in Construction*, 127, 103705. (Q1 journal).

Ekanayake, B., Fini, A. A. F., and Wong, J. K. W. (2021), Technical challenges for automated indoor construction progress monitoring. *44th AUBEA Conference*, Deakin University, Melbourne, Australia, 28–29 October 2021, pp 152–162.

Ekanayake, B., Fini, A. A. F., Wong, J. K. W., and Smith, P. (2022), A deep learning based approach to facilitate the as-built state recognition of indoor construction. *Construction Innovation*. (Q1 journal).

Abstract

Inefficient and inaccurate monitoring of work-in-progress significantly contributes to schedule and budget overruns in construction projects. The adoption of computer vision technologies has expedited automation, accuracy, and efficiency in the process of monitoring construction progress by overcoming the challenges of laborious, costly and error prone manual methods. The majority of computer vision based progress monitoring studies have focused on the exterior construction environment and few studies have been conducted for indoor sites. The literature review found that robust object recognition from indoor site images is inhibited by technical challenges related to indoor objects, lighting conditions and camera positioning. Compared to conventional machine learning algorithms, deep learning models are robust in automated visual recognition of as-built elements and calculation of work completion. Therefore, this study aimed to develop a deep learning based approach for improving automation in computer vision based indoor construction progress monitoring. The design science research framework was employed as the underpinning methodology since this study focused on addressing a practical problem by developing a technology based artefact.

Three indoor sites were used as case projects and time-lapse cameras were used to capture progress images. The framing, insulation and drywall installation of indoor walls was selected as the progress monitoring scenario. The images captured were first used to demonstrate the technical challenges and their impacts on feature extraction. Using transfer learning and hyperparameter fine-tuning, a Mask Recurrent Convolutional Neural Network (Mask R-CNN) model was optimised. Google Colab was used to train the deep learning model on the images captured to reduce computational complexities. It was found that visual recognition of indoor as-built elements of the Mask R-CNN model was improved by optimising the learning rate with Adam optimiser and cosine annealing. Training efficiency improvements were made through the batch size, network resolution and number of iterations. The model reported a mean average precision of 88.02%, and training and validation losses of 0.19 and 0.20 respectively. This model was post-processed to automatically calculate the work-in-progress of indoor elements by extracting the areas of the segmented pixel masks of the as-built states detected.

The key contribution of this research is the development of an optimised and post-processed model of Mask R-CNN, which is capable of automated as-built visual

recognition and work-in-progress measurement of indoor construction elements. This study also contributes to offering a simplified, efficient, and shareable training workflow on Colab that can be replicated in similar studies. The novelty of this research is in the optimisation process of the Mask R-CNN model and the Colab based training workflow. Upon providing training images, the Mask R-CNN model and the training workflow can be extended to other as-built scenes to improve automation in the progress monitoring process. In order to achieve this, the study provides an easy-to-follow technical flow diagram and further explanations on the workflow of the deep learning model, which can be understood by construction professionals with a non-programming background.

1 Introduction

1.1 Background

Building construction projects are usually completed with schedule and cost discrepancies (Love et al., 2005; Pal et al., 2022). K. Han et al. (2015) claim that more than 53% of building construction projects have delays and more than 66% of projects encounter budget overruns. Inefficient and inaccurate monitoring of work-in-progress is one of the major factors that lead to project schedule and budget overruns (Omar et al., 2018). Construction progress monitoring focuses on collecting, analysing, reporting, and visualising the progress information of construction elements, activities and operations (Bohn & Teizer, 2010; Golparvar-Fard et al., 2015). Real-time and accurate monitoring of construction elements ensures project cost and schedule control, status reporting, resource management, productivity measurement, efficient cashflow, progress payments and claims reporting (Yang et al., 2015; Zhang et al., 2009).

Traditional methods of progress monitoring require manual extraction of progress data from site drawings, schedules, and daily site reports generated by different construction personnel (Golparvar-Fard et al., 2015). The studies of Golparvar-Fard et al. (2015) and Zhang et al. (2009) point out that traditional practices of as-built state recognition and work-in-progress calculation of construction elements employ visual observations, tape and red pencil based approximate assessments, which lead to creating misleading information. The traditional practices are labour intensive, time consuming, costly, visually complex, and prone to errors (Bosché, 2012; Golparvar-Fard et al., 2015; Yang et al., 2015). Improving automation in construction progress monitoring enables real-time data collection, accurate information extraction with minimum human bias and visually friendly reporting to be effectively used by all project participants (Kopsida et al., 2015). The state-of-the-art technologies such as, images collected by cameras, point clouds generated by laser scanning, radio wave based technologies and geospatial based technologies (Omar & Nehdi, 2016) have contributed to improving automation in progress monitoring process. They are mainly integrated with building information models (BIM) for progress monitoring purposes (Kopsida et al., 2015). The use of unmanned aerial vehicles (UAV) such as drones and unmanned ground vehicles (UGV) such as robot dogs (Ibrahim et al., 2019; McCabe et al., 2017) have also contributed to improving automation in progress data collection. The real-time monitoring at the level

of the element demonstrates a dynamic evolution of the project (Martinez et al., 2019; Z. Wang et al., 2021). However, existing studies generally focus on monitoring the progress of externally observable construction works such as columns, beams, and walls (Golparvar-Fard et al., 2015; Han & Golparvar-Fard, 2017).

Because of the major focus of automated progress monitoring on the exterior construction environment, indoor construction progress monitoring has not been fully explored (Deng et al., 2020; Hamledari & McCabe, 2016; Kropp et al., 2018; Moragane et al., 2022). The interior mainly includes elements such as building services, partitions, finishes and fittings (Hamledari & McCabe, 2016; Hamledari, McCabe, & Davari, 2017). Due to the cluttered and complex layout, construction processes become more complicated in indoor sites (Hamledari & McCabe, 2016; Kropp et al., 2013). Any delay to interior works affects the succeeding activities, thus delaying the overall project completion (Gurevich & Sacks, 2014). Since the average budget of the interior works is about 25% to 40% of the entire project budget, budget overruns related to indoor works affect the overall project budget (Kropp et al., 2012). Therefore, real-time efficient and accurate monitoring of the work-in-progress of interior works is crucial (Hamledari, McCabe, & Davari, 2017; Kropp et al., 2018).

Unlike an exterior construction environment, the indoor construction environment poses challenges for progress data collection and information extraction (Shahi et al., 2013). Because of the blockage of satellite signals indoors, capturing global positioning system (GPS) based data is challenging (Ibrahim et al., 2019). The radio wave based technologies such as ultra-wide band (UWB) devices are affected as signals are blocked by indoor materials (Ham et al., 2016). Laser scanners generate less accurate point clouds for reflective indoor materials such as metal studs and pipes (Golparvar-Fard, Bohn et al., 2011; Turkan et al., 2012). Continuous advances in digital cameras and image processing algorithms have enhanced accuracy and reliability of computer vision (Bohn & Teizer, 2010; Deng et al., 2020) for indoor construction progress monitoring (Hamledari & McCabe, 2016; Hamledari, McCabe, & Davari, 2017). The competitive advantages of computer vision can be realised in cost, time, and convenience of use for progress data collection and processing (Bohn & Teizer, 2010; Deng et al., 2020; Ham et al., 2016).

Computer vision based progress monitoring entails acquiring, processing, and understanding 2D and 3D site images with the use of cameras and machine learning

algorithms (Ekanayake, 2022; Szeliski, 2010; Zhang et al., 2009). There have been a growing number of computer vision based indoor construction progress monitoring studies (Deng et al., 2020; Hamledari, McCabe, & Davari, 2017; Hamledari, McCabe, Davari, Shahi et al., 2017; Kropp et al., 2014; Kropp et al., 2018). Using machine learning algorithms, these studies have exploited the handcrafted extraction of features such as edges, colours, and texture (O'Mahony et al., 2019) to recognise indoor elements from the progress images. Some existing studies have observed that the automated visual recognition of the as-built state is challenging in the indoor environment due to reasons such as occlusions, achromatic indoor objects, cluttered indoor scenes, and lighting conditions (Deng et al., 2020; Hamledari & McCabe, 2016; Kropp et al., 2014). The machine learning algorithms relying on handcrafted features are sensitive to the visual quality of input images and are difficult to extend to new image datasets (Ying & Lee, 2019). The region of interest in the images of poor visual quality cannot be detected easily without initially performing pre-processing algorithms to remove background noise and lighting impacts (Nanni et al., 2017). Although the existing studies on computer vision based indoor construction progress monitoring have discussed these technical challenges, the current understanding of the challenges is still poor. The existing studies have not adequately demonstrated the impacts of the technical challenges for recognition of indoor as-built elements. Having a better understanding of challenges will lead to creating more robust customised algorithmic solutions for the automated recognition of indoor as-built scenes.

With the advances in computer vision, the use of deep learning to devise algorithmic solutions is emerging in construction progress monitoring studies (Akinosho et al., 2020; Ekanayake, 2022; Z. Wang et al., 2021). Deep learning is a branch of machine learning. Deep learning models learn features by training large amounts of data, unlike typical machine learning models (Chollet, 2017; LeCun et al., 2015). This learning ability of deep learning models through training images enables a single object recognition algorithm to be used to detect and classify objects, without implementing additional steps of pre-processing (Ying & Lee, 2019). The advances in deep learning have contributed to recognising more complex objects in indoor environments with decent accuracy (B. Wang et al., 2022). Therefore, deep learning approaches not only improve automation but also reduce the inaccuracies caused by programmer's biases in manual feature extraction (Slaton et al., 2020). The recent developments in computer vision based indoor

construction progress monitoring have harnessed the detection, and classification ability of deep learning models to recognise indoor elements with irregular shapes (Shamsollahi et al., 2021; Ying & Lee, 2019). The object recognition accuracy of the deep learning model can be improved through algorithmic optimisations (W. Wei et al., 2022) to minimise the impacts of technical challenges. However, building and training an optimised deep learning model is computationally complex (O'Mahony et al., 2019). Therefore, there is a need to optimise a deep learning model for robust visual recognition by reducing the high computational resource requirements and training configuration complexities. The optimised deep learning model will contribute to robust automated visual recognition of indoor as-built scenes upon providing training images.

In addition to automated visual recognition, existing studies on computer vision based indoor construction progress monitoring have not been able to fully automate the work-in-progress calculation process. Hamledari, McCabe, Davari and Shahi (2017) proposed a machine learning approach to automatically detect and classify the components of an indoor partition. However, this algorithm is not able to calculate the work completion percentages automatically without being subject to project manager's opinions. Kropp et al. (2018) developed an interior construction state recognition machine learning approach with BIM registered images to update schedule deviations. This method is only applicable when the construction element is fully completed. Even though the handcrafted machine learning algorithms used in previous studies classify the state of the construction elements, they fail to automatically calculate the percentage work completion. The segmentation ability of deep learning has been explored for calculating the indoor work-in-progress in W. Wei et al. (2022). However, their study has not demonstrated the area calculations of the segmented masks from the deep learning model when a progress image is provided. With the application of the optimised deep learning model, which uses segmentation in addition to detection and classification, there is potential to facilitate the automation in work-in-progress calculation of indoor elements. This solution will contribute to automatically calculating the work completion percentages of an indoor as-built scenario on a given day when an indoor site image is provided.

1.2 Problem statement

With the deficiencies in the current computer vision based indoor construction progress monitoring research, three research gaps identified for the current study are summarised as follows. First, the current understanding of the technical challenges in indoor sites and

their impacts that impede robust visual recognition need to be investigated to devise customised algorithmic solutions. The current solutions of automated visual recognition for indoor elements depend heavily on handcrafted machine learning algorithms. Harnessing the detection and classification ability of deep learning algorithms can lead to robust solutions by minimising the impacts of technical challenges. Despite the capabilities of deep learning to improve automation in visual recognition of as-built indoor elements, application of deep learning algorithms seems to be lacking for indoor construction progress monitoring due to the computational complexities of deep learning models. The computational complexities of deep learning are mainly two-fold: the development of the model and training of the model. Second, there is a need to optimise a deep learning model for robust visual recognition of indoor as-built elements by reducing the computational complexities. Third, with the application of the optimised deep learning model, there is a need to facilitate automated calculation of work-in-progress. By recapitulating the gaps in existing studies, the central research problem of the present study has been identified as **how to improve automation in computer vision based indoor construction progress monitoring**.

1.3 Research questions

Based on the research gaps, the following three research questions have been formulated.

1. What are the technical challenges in indoor construction environments and how do they affect the automated visual recognition of as-built elements?
2. How can deep learning be employed for automated visual recognition of indoor as-built elements by reducing the computational complexities?
3. How can deep learning be employed to facilitate the automated calculation of work-in-progress of indoor elements?

1.4 Research aim and objectives

The aim of this study is **to develop a deep learning based approach for improving automation in computer vision based indoor construction progress monitoring**. To pursue this aim, the following objectives have been outlined.

1. Investigating the technical challenges and their impacts for indoor as-built elements visual recognition.

2. Developing an optimised deep learning algorithm for automated visual recognition of as-built indoor elements by reducing the computational complexities.
3. Automatically calculating the work-in-progress of indoor elements by post-processing the optimised deep learning algorithm.

1.5 Overview of the research methodology

An overview of the methodology used to attain the research objectives is presented in this section. Detailed accounts and justifications related to research design and choice of methods are explained in Chapter 3. Initially, an extensive literature synthesis was conducted to review current progress monitoring in construction projects and to present a review of existing studies on computer vision based indoor construction progress monitoring. The research gaps were identified, and the research problem, questions, aim, and objectives were formulated. Through the literature review, the technical challenges for computer vision based as-built indoor elements visual recognition were identified. To corroborate and modify the literature findings and to demonstrate the impacts of technical challenges, empirical evidence in the form of progress images was gathered from three case projects. An optimised Mask Recurrent Convolutional Neural Network (Mask R-CNN) was employed as the deep learning model for automated visual recognition of indoor as-built elements. The optimised Mask R-CNN model was post-processed to monitor and measure the work completion of indoor elements. The case projects are used only for demonstration purposes. If high quality images are collected, the nature of the indoor construction activity has minimal impact on the deep learning model. The process of framing, insulation, and drywall installation of indoor walls was selected as the progress monitoring scenario to demonstrate the performance of the optimised deep learning model. Of the three case projects, two sites were used to capture training images while the third site was used to collect test images. Time-lapse cameras were installed to capture image data.

1.6 Scope and limitations

Indoor walls with the process of framing, insulation, and drywall installation were selected as the progress monitoring scenario to demonstrate the performance of the optimised deep learning model. Indoor walls are a significant part of interior construction and delays related to them can have costly consequences (Kropp et al., 2012). The

progress data capturing, and information extraction related to indoor walls can aid many trades such as site managers, framers, insulation installers and drywall installers. To infer the as-built state and work-in-progress for an indoor wall, information related to installation of components such as frames, insulation blankets and plasterboards is required (Hamledari, McCabe, & Davari, 2017).

Although BIM contributes to as-planned modelling of the construction site and visualising the progress deviations efficiently, some construction stages are not usually modelled in BIM (Pal et al., 2022). This is particularly applicable for indoor construction activities (W. Wei et al., 2022; Ying & Lee, 2019). In practice, detailed BIMs are not widely generated for small and medium scale fit-out, renovation and refurbishment projects. Overall, half of Australian small and medium scale contractors employ computer aided design (CAD) models (Hong et al., 2019) and the use of BIM is still not very popular among Australian building contractors (Ahankoob et al., 2021). The current study only focuses on as-built visual recognition and calculation of the work-in-progress using the optimised deep learning model when a prior model such as BIM is not available. Upon providing training images, the optimised deep learning model can be extended to other indoor as-built scenes.

This PhD study was undertaken during the global pandemic of COVID-19. The COVID-19 pandemic created unique challenges in construction projects (Oo & Lim, 2021). This research was affected by the challenges that emerged from COVID related restrictions. During the data collection period of April 2021 to May 2022, the selected case projects were heavily affected by COVID lockdowns and restrictions imposed in the state of New South Wales (NSW), Australia. As a result of COVID outbreaks, construction activities across NSW were temporarily shut down on 19 July 2021 and resumed work for vaccinated workers from 11 August 2021. The stay-at-home COVID restrictions for NSW residents were only lifted on 11 October 2021 allowing visitors to access construction sites (Master Builders Association New South Wales, 2021). As a result, only two indoor construction projects were used to collect training images and the third project was used to collect test images. With the COVID outbreaks, construction personnel being infected and material supply issues, the construction works in case project 2 and case project 3 were delayed. The scope of the study was thus limited to capturing the training and testing images of one progress monitoring scenario due to the time constraints imposed by COVID restrictions. The use of fixed time-lapse cameras to capture the progress

monitoring scenario ensured that efficient and accurate site image collection was undertaken abiding by the COVID related social distancing rules.

1.7 Outline of the thesis

This thesis is organised in six chapters as briefly explained below. Mapping of the research objectives, and methods corresponding to the chapters is presented in Figure 1.1.

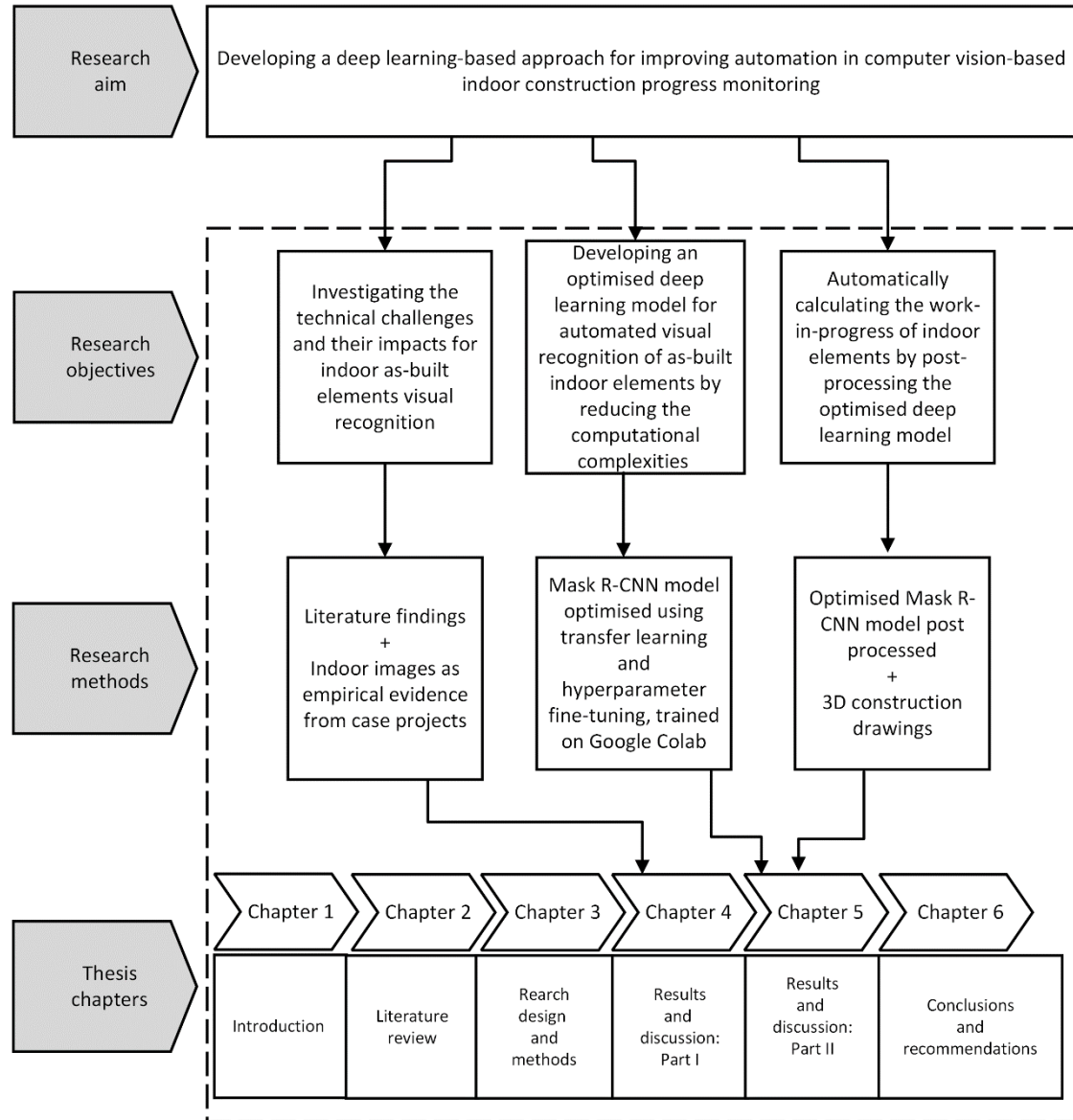


Figure 1.1: Mapping of research objectives, methods and corresponding chapters

Chapter 1: Introduction

This chapter provides the background to the study and formulates the research problem, gaps, questions, aim and objectives. It also provides an overview of the research methodology and discusses the scope and limitations.

Chapter 2: Literature review

This chapter reviews the construction progress monitoring methods with an emphasis on the challenges of traditional methods and advances in state-of-the-art technologies. The chapter discusses the algorithmic approaches to analyse the as-built state highlighting the use of computer vision based technologies spanning from machine learning models to deep learning models. The chapter also reviews the advances in computer vision based indoor construction progress monitoring. The research gaps stemming from the existing computer vision based indoor construction progress monitoring are reviewed at the end of this chapter.

Chapter 3: Research design and methods

Chapter 3 determines the methodological framework of this research. This chapter explores the data collection process, case projects and creation of the training images dataset. In this chapter, the Mask R-CNN model and its development, training and evaluation process is briefly reviewed.

Chapter 4: Results and discussion: Part I

Chapter 4 presents and discusses the empirical evidence from the case projects corroborating the literature findings on technical challenges and their impacts on as-built indoor elements visual recognition. The findings discussed in this chapter contribute to fulfilling Objective 1 of this research.

Chapter 5: Results and discussion: Part II

This chapter has two major parts. In the first part, the development, training and hyperparameter fine-tuning process of the Mask R-CNN model for automated visual recognition of indoor elements is presented and discussed to achieve Objective 2. The second part presents and discusses the workflow for calculating the work-in-progress of indoor elements using the post-processed optimised Mask R-CNN model. Objective 3 of this research is achieved by using the results.

Chapter 6: Conclusions and recommendations

To conclude the thesis, this chapter first summarises the achievement of the research objectives and the outcomes, then discusses the theoretical and practical contributions of this study. After discussing the limitations of the current study, this chapter offers the recommendations for future studies.

2 Literature review

This chapter presents a review of progress monitoring in construction projects in section 2.1. After highlighting the challenges of traditional methods, the chapter provides an overview of the state-of-the-art technologies in section 2.2 and of algorithmic approaches applied to construction progress monitoring in section 2.3. The emphasis is on computer vision based technologies. The chapter also highlights the paucity of studies on indoor construction progress monitoring in section 2.4, while noting the lack of deep learning based approaches to improve automation. The three research gaps identified in this study are extensively reviewed in section 2.5. Section 2.6 provides the chapter summary.

2.1 Challenges of traditional construction progress monitoring

Construction progress monitoring is interchangeably used with the terminologies progress measurement, assessment, and tracking. It is interpreted as collecting, analysing, reporting, and visualising the progress information of construction elements (Golparvar-Fard, Bohn et al., 2011). It is crucial for project managers to understand, in a timely manner and accurately, how a project is progressing (Alizadehsalehi & Yitmen, 2018; Kopsida et al., 2015). Construction progress monitoring occurs at the operational level and project level. While operational level monitoring captures the causes of progress deviations occurring on a daily basis, project level monitoring measures the extent of adherence to the plans in constructing major building elements of the project (Yang et al., 2015). For monitoring a project in terms of both these levels, accurate and real-time capture, analysis, and visualisation of the as-built work-in-progress calculation are crucial (K. Han et al., 2015). Documenting the progressive as-built state and measuring the level of work completed during construction works (Hamledari et al., 2018) are mainly conducted for the purposes of facilitating progress payment, determining deviations from the schedule, and taking remedial action to address any budgetary and delay issues (Kropp et al., 2014).

Monitoring of construction progress heavily relies on traditional manual methods (Ham et al., 2016). Traditional construction progress monitoring is predominantly conducted by filling out daily site reports and through manual inspections (W. Wei et al., 2022). Daily site reports hold data, which is manually logged and detailed on a daily basis (L.Y. Liu et al., 1995). The reports are used to obtain information related to work and problems encountered on a daily basis (Abdelsayed & Navon, 1999). Manual inspections for

progress data collection entail walkthrough assessments by site managers and site supervisors (Ham et al., 2016). The traditional practices of as-built state recognition and work completion measurement of construction elements rely on approximate visual assessments taken using tape and red pencil (Golparvar-Fard et al., 2015; Zhang et al., 2009). It is evident that traditional construction progress monitoring encompasses manual methods of data collection, information interpretation, measurement, reporting and visualisation (Golparvar-Fard et al., 2015). The reliability of the progress evaluation tools and reports depends predominantly on the accurate and timely collection of site data (El-Omari & Moselhi, 2011). Information derived from site data tends to be subjective, based on construction personnel's interpretations (Golparvar-Fard et al., 2015). Moreover, improper understanding of the as-built state can lead to erroneous decision making by the project team (Kopsida et al., 2015). K. Han et al. (2015) explained that the contractors, subcontractors, and clients can have disagreements about the project progress on a given date because of the inaccuracies and inefficiencies in traditional progress monitoring. Even if the progress is measured, it may be performed in an ad-hoc manner and the interpretations could be subjective (Golparvar-Fard et al., 2015) and lack the required accuracy (Omar & Nehdi, 2016).

Traditional paper based progress reports are visually intricate and difficult to analyse at project meetings, attended by tradespeople with various areas of expertise and interests (K. Lee et al., 2006). These progress reports may not reflect the actual circumstances of the project (Golparvar-Fard, Bohn et al., 2011; Navon, 2007). As a result, additional time needs to be spent on explaining the context in which problems occur rather than understanding the causes of the problems. The majority of time in meetings is spent on descriptive (35%) and explanative (42%) tasks and only 12% and 11% of the time is spent on evaluative and predictive tasks respectively (Golparvar-Fard, Bohn et al., 2011). Since traditional representations of information do not provide an accurate picture of the as-built conditions (Koo & Fischer, 2000; Poku & Arditi, 2006), disputes among the project participants and flawed project performance are inevitable (K. Han et al., 2015).

In progress monitoring approaches, it is crucial to reduce errors, and facilitate appropriate and timely corrective actions. Inefficiencies and inaccuracies in traditional progress monitoring methods are among the contributing factors to building construction projects encountering overruns of 20% in the schedule and overruns of up to 80% in the budget (Agarwal et al., 2016). Improving automation in construction progress monitoring

enables real-time and accurate data collection, information processing with minimum human bias, and efficient reporting (Kopsida et al., 2015).

2.2 State-of-the-art progress monitoring approaches

Automated systems with state-of-the-art technologies could be designed with less human intervention and human bias in the overall system (Zhang et al., 2009). The adoption of digital technologies has enhanced the performance of construction projects with increased levels of automation. Computer simulation, modelling and computer vision approaches have largely been used during the last decade to address the growing need to improve automation in project progress monitoring (Reja et al., 2022; Shafieezadeh-Abadeh et al., 2019). Nevertheless, achieving full automation in construction progress monitoring is not possible since human intervention is essential to establish the connections between work components, determine design changes and make decisions (Zhang et al., 2009).

The automated progress monitoring process can be divided into the following steps as suggested by Kopsida et al. (2015) and Omar and Nehdi (2016): (a) data acquisition, which refers to capturing as-built scenes, (b) information extraction, which involves the process of extracting the information needed from the as-built data, (c) progress estimation which includes comparison between the as-built and as-planned model to calculate the work-in-progress, and (d) visualisation of the results. This process is illustrated in Figure 2.1.

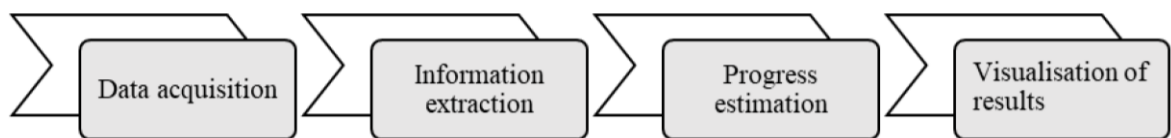


Figure 2.1: Process of automated progress monitoring in construction projects

Adapted from: Kopsida et al. (2015) and Omar and Nehdi (2016)

Although the early efforts of automating progress monitoring were evident in comparing the captured as-built images with the project schedule, determining the progress deviations were still subjected to the judgements of inspection personnel (Zhang et al., 2009). Introducing evidence for visual inspection and reducing the number of site visits are the automation aspects of capturing progress using digital images (Fard & Peña-Mora, 2007; Zhang et al., 2009). The ubiquitous use of digital cameras in construction sites

mainly for security purposes (Ahmadian Fard Fini et al., 2022; Wong et al., 2022) is a contributing factor in the growth of site images based visual inspection (Hamledari, McCabe, & Davari, 2017). The site images based visual inspection practices contributed to the adoption of computer vision in the process of monitoring construction progress (Kopsida et al., 2015; Omar & Nehdi, 2016). Early developments in progress monitoring systems focused on manual comparison of the site images with 2D drawings and construction schedules (Zhang et al., 2009).

Apart from computer vision, the 3D reconstructions through point clouds obtained from laser scanning are used for capturing volumetric data (Bosché et al., 2015; Shahi et al., 2013). Another type of data collection system is radio wave based object identification which includes barcoding, radio frequency identification (RFID) and ultra-wide band (UWB) technologies to track position data (Hubbard et al., 2015; Omar & Nehdi, 2016). Moreover, geospatial based technologies such as global positioning systems (GPS) are also employed to capture position data (El-Omari & Moselhi, 2011; Kang & Tesar, 2004). These advanced data acquisition technologies are used for real-time construction progress monitoring with the integration of 3D computer aided design (CAD) (Bosché, 2010) and BIM (Bosché et al., 2015) to track the project progress more efficiently. Employing 3D CAD as the as-planned model integrated with 3D point clouds of the as-built scenes (Bosché, 2010; Bosche & Haas, 2008; Bosche et al., 2009) is one example. In the past decade, computer vision based as-built models and 3D point clouds have been linked with BIM to simulate construction schedules in the fourth dimension or 4D (3D + Time) BIM (Golparvar-Fard et al., 2015; Han & Golparvar-Fard, 2017; Park et al., 2018). The adoption of 4D BIM in conjunction with machine learning algorithms to extract progress information has been promising (Braun & Borrmann, 2019; Golparvar-Fard et al., 2015).

Despite the advantages of the state-of-the-art technologies compared to traditional methods, there is opportunity for more improvements in automation, accuracy, and efficiency in the progress monitoring process (Z. Wang et al., 2021). Various raw data items such as images, laser scans, positions and geo tags of the construction elements are captured and processed using these technologies (Lu & Lee, 2017). The following subsections explore how the prominent data acquisition sensors, namely laser scanning, radio wave based technologies, geospatial based technologies, and computer vision based cameras, have been employed in the construction progress monitoring process, which is shown in Figure 2.1.

2.2.1 Laser scanning based technologies

Amongst the state-of-the-art technologies, laser scanning is the most efficient method to create the 3D view of construction objects (M.-K. Kim et al., 2016). 3D laser scanning uses the Light Detection and Ranging (LiDAR) mechanism. A laser scanner calculates the distance to a target object by emitting laser beams and sensing the reflected signals from the target object. This process in turn creates a point cloud (Turkan et al., 2012), which is a 3D reconstruction of the target object. The time-of-flight, and the phase shift are the main two techniques for range measurement. Scanners, which employ the time-of-flight technique produce a laser pulse and calculate the travelling time of the reflected pulse. Scanners that utilise the phase shift technique produce an amplitude modulated continuous wave and calculate the phase shift between the emitted and reflected signals (Lee & Park, 2019). 3D laser scanners can also be categorised based on the mounting techniques: terrestrial laser scanner (TLS), airborne laser scanner (ALS), and mobile laser scanner (MLS). A typical terrestrial laser scanner is shown in Figure 2.2.



Figure 2.2: Trimble GX 3D laser scanner

Source: Turkan et al. (2012)

Utilising of laser scanners for site progress monitoring is evident in early studies, which used a 3D CAD as the as-planned model and 3D point clouds as the as-built model (Bosché, 2010; Bosche & Haas, 2008; Bosche et al., 2009). 3D laser generated point clouds are increasingly in use with the integration of BIM models to determine progress deviations and update the schedule for major construction elements such as columns and, walls (Bosché et al., 2014; M.-K. Kim et al., 2016; Shirowzhan et al., 2018; Turkan et al.,

2013; Yang et al., 2020). Generating an as-built model with a point cloud involves the steps of point cloud registration, noise filtering, removal of outliers, and down-sampling (Reja et al., 2022) with a high degree of manual intervention to ensure the quality and accuracy of the point cloud based as-built model (Q. Wang et al., 2020). Registration entails combining multiple partial point clouds into a single file or superimposing a single as-built point cloud over the as-planned BIM model. The unwanted points such as noise and outliers are then removed to enhance the point cloud quality. Since the integration of point clouds causes the overlapped regions to become denser, down-sampling is performed to create a uniform point cloud (Reja et al., 2022; Q. Wang & Kim, 2019). The 3D reconstruction of a building is presented in Figure 2.3 to demonstrate the point cloud based as-built model.



Figure 2.3: 3D as-built reconstruction of a building using point cloud

Source: Xiong et al. (2013)

The point cloud is registered over the as-planned BIM, and the as-built elements are usually detected based on overlap occupancy to compare the as-built and as-planned progress (Q. Wang et al., 2020). Many studies use traffic light based coding to illustrate the different levels of work-in-progress in terms of red for not completed, yellow for in progress and green for fully completed state (Golparvar-Fard, Peña-Mora et al., 2011). Advances in gaming engines have contributed to integration of point clouds and BIM with augmented and virtual reality environments (Zollmann et al., 2014) to visualise the progress deviations. The merits and drawbacks of laser scanning technology for progress monitoring studies are discussed in Table 2.1.

2.2.2 Radio wave based technologies

Radio wave based technologies use radio frequencies for capturing and transmitting data from memory chips, which are usually referred to as tags. They are embedded or attached to the object of interest (Omar & Nehdi, 2016). They can store and retrieve data from the

object of interest and this information is easily read and written (Ergen & Akinici, 2007; Ergen et al., 2007). Commonly used radio wave based techniques are barcoding (Navon & Sacks, 2007), RFID (Omar & Nehdi, 2016) and UWB (Cho & Kang, 2010).

Barcoding technology incorporates barcodes, which are a sequence of parallel and adjacent bars. They are scanned with a barcode reader, either handheld or stationary, to capture information (Oh et al., 2004). The barcode technology is primarily used for construction material tracking and inventory monitoring (Navon & Sacks, 2007). RFID refers to a category of automated identification technologies in which radio frequencies are used to transmit data from memory chips attached to objects. RFID readers are either fixed, handheld or vehicle mounted (Navon & Sacks, 2007). RFID tags are widely used for material tracking in construction sites (Shahi et al., 2015). RFID tags, barcode tags and handheld readers are shown in Figure 2.4.



Figure 2.4: RFID and barcode technologies

Source: T.-Y. Lin et al. (2014)

Similar to RFID, UWB is a radio based technology that is used for short range communications (Cheng et al., 2011). The tag attached to the target object transmits UWB radio pulses that facilitate the system to locate its 3D position coordinates (Cho & Kang, 2010). Some examples of employing UWB technology include construction material tracking and activity based progress tracking for safety related applications (Shahi et al., 2013).

Although the radio wave based technologies have not been directly used for physical progress measurement and schedule updating, they are employed for material tracking

(Hubbard et al., 2015; Shahi et al., 2015). The efficient material tracking ability has been used in conjunction with laser scanning which generates 3D point clouds of the as-built scenes (Valero et al., 2012; Valero et al., 2016). Radio wave based technologies lose their effectiveness when the work moves indoors (Razavi & Haas, 2011). Valero et al. (2012) and Valero et al. (2016) have demonstrated how RFID tags can be used in conjunction with point clouds in furnished interiors for material tracking and volumetric progress data capturing in progress monitoring systems. Therefore, integration of the technologies for progress monitoring in construction sites alleviates shortcomings associated with each technology when employed individually (Maalek & Sadeghpour, 2013; Razavi & Haas, 2010; Valero et al., 2012). The merits and drawbacks of radio wave based technology for progress monitoring studies are discussed in Table 2.1.

2.2.3 Geospatial based technologies

The navigation systems that provide 3D positions use satellite-based technologies such as GPS. They require a line-of-sight between the receiver and the satellite. The functional ability of these technologies relies on receiving signals from satellites to locate the position of the object of interest, when they are employed in progress tracking in construction sites in terms of geographic location (El-Omari & Moselhi, 2011; Kang & Tesar, 2004). GPS data loggers are generally used to collect tracking data in construction sites (Pradhananga & Teizer, 2013). A GPS data logger is shown in Figure 2.5. Geographic information systems (GIS) are useful tools to understand the geographic context captured from the GPS data. A GIS comprises of digital geographic data, and tables of attributes connected to map identities to visualise the data in a mapped format (Bansal & Pal, 2009).



Figure 2.5: GPS data logger

Source: Pradhananga and Teizer (2013)

Behnam et al. (2016) introduced a prototype system, which automatically generates and displays the planned and actual progress of construction tasks in linear infrastructure projects such as a railway using satellite remote sensing techniques and a GIS web based platform. The system proposed by Behnam et al. (2016) incorporates the integration of multi sensor data since it uses satellite images and 2D site images. The applications of integrating two or more sensor technologies have been explored in a few GPS and GIS based studies. Although GPS technology fails to determine the signals from indoor locations, recent developments have facilitated GPS to operate indoors by integrating with laser scanners (Kang & Tesar, 2004). A 3D Web GIS system was established to execute the algorithm and show the location of the object of interest indoors (Ko, 2013). The merits and drawbacks of geospatial based technology for progress monitoring studies are discussed in Table 2.1.

2.2.4 Computer vision based technologies

Among the state-of-the-art technologies, the adoption of computer vision for construction progress monitoring fundamentally entails replicating human vision using images captured from cameras and analysing images using image processing algorithms (Ekanayake, 2022; Zhang et al., 2009). Applying computer vision to process and analyse images, detect, measure, and track civil infrastructure construction has attracted both civil engineers and computer scientists (Chi & Caldas, 2011; Yang et al., 2015).

Amongst the computer vision based data capturing tools, image acquisition using digital cameras, which produce 2D images, are prominent in site data collection (Szeliski, 2010). These cameras are high definition (HD), which can capture footage of resolution 1280 x 720, i.e., 720 pixels, or 1920 x 1080, i.e., 1080 pixels (Tamayama et al., 2002). The intrinsic parameters (focal length, principal point, lens distortion coefficient) and the extrinsic parameters (the relative positions of the grids with respect to the camera) of the cameras should be taken into consideration when capturing images (Fetić et al., 2012). In terms of 2D imaging, monocular images and stereo images are generated corresponding to monocular and binocular cameras, and video cameras respectively (Ma & Liu, 2018). One of the most popular and economical means of collecting progress images is using fixed time-lapse HD cameras. In time-lapse photography, the frequency at which the frames are captured is lower than the frequency used to view the sequence. Therefore, when a video captured from a time-lapse camera is played at normal speed, time appears to be moving faster. This is referred to as lapsing (Yang et al., 2015). The constraints related to limited coverage and range of cameras can be remedied by using other camera mounting techniques. Unmanned aerial vehicles (UAV) such as drones and unmanned ground vehicles (UGV) such as robot dogs (Ibrahim et al., 2019; McCabe et al., 2017) are key examples. Range images or the depth images are another category of digital images, which are captured using red, green, and blue colour channels with depth information (RGB-D) (Omar & Nehdi, 2016). This information is used to map and create dense 3D reconstruction of buildings and construction elements. Images are acquired with range sensors or range cameras (Omar & Nehdi, 2016; Reja et al., 2022). Smart phones and personal digital assistant (PDA) tablets are increasingly used as handheld devices for capturing site images (Omar et al., 2018).

In the early construction progress monitoring approaches, randomly collected daily construction images were used to capture the as-built status (Fard & Peña-Mora, 2007; Zhang et al., 2009). With advances in computer vision, 3D as-built image models are built from photogrammetry techniques such as structure-from-motion (SfM) and simultaneous localisation and mapping (SLAM) by converting 2D images into 3D reconstructions (Amer et al., 2018). These images are either compared with each other or superimposed over a 4D BIM to track schedule discrepancies (Amer & Golparvar-Fard, 2018a; Golparvar-Fard et al., 2015). Apart from the algorithmic approaches of SfM and SLAM, commercially available photogrammetry software is employed to generate 3D

reconstructions (Reja et al., 2022). The SfM based 3D as-built model superimposed over 4D BIM for determining progress deviations is evident in the studies of Braun and Borrmann (2019), Braun et al. (2020) and Golparvar-Fard et al. (2015) for exterior building elements such as columns, beams, formwork, and reinforced walls. Omar et al. (2018) used commercial photogrammetric software to create a 3D as-built model from 2D images, which they referred to as a point cloud of building columns registered with the as-planned BIM. This progress monitoring system was integrated with an automatic email and text message based notification system to inform the project manager of any progress deviations in constructing columns (Omar et al., 2018). The construction site in a 2D image and 3D photogrammetric model is shown in Figure 2.6.



Figure 2.6: A construction site a) 2D image; b) 3D photogrammetric model

Source: Omar et al. (2018)

Although photogrammetric techniques are efficient in creating 3D models and linking with the BIM, this method requires a meticulous manner of capturing images to understand the 3D view of each construction scene. This can be time consuming and requires manual intervention. Additionally, the BIM must have thorough detailed elements modelled before being superimposed (Golparvar-Fard et al., 2015; Omar et al., 2018). In another branch of studies, through machine learning based image processing, information is extracted from 2D site images and integrated with a 4D BIM (Braun & Borrmann, 2019; C. Kim et al., 2013; Park et al., 2018). An example is presented in Figure 2.7.

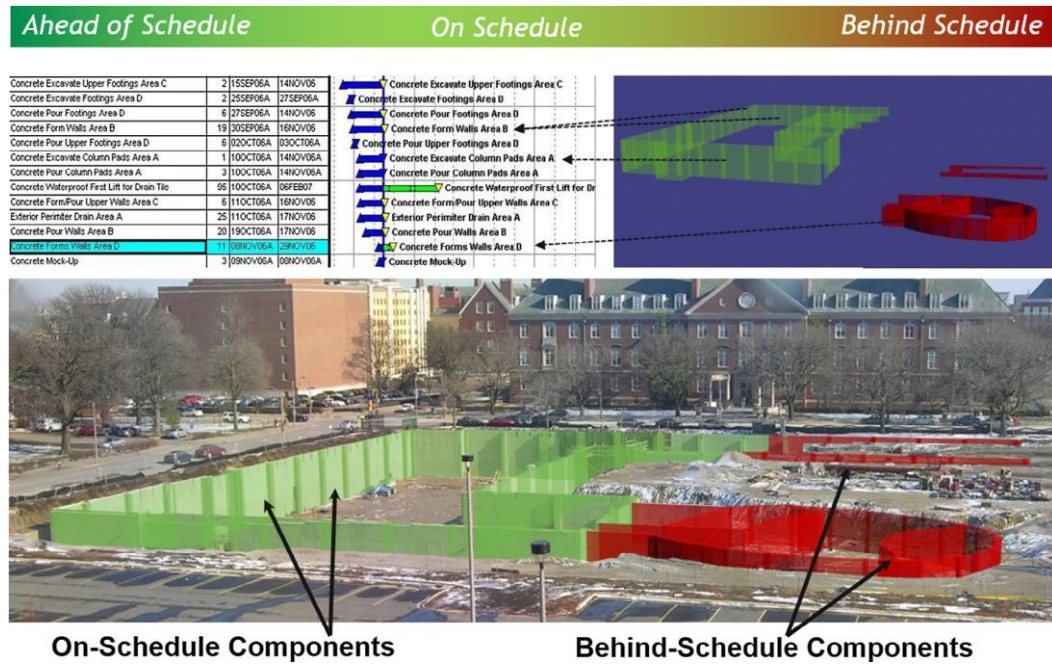


Figure 2.7: 2D site images integrated with 4D BIM

Source: Yang et al. (2015)

Generally, the details of the superstructure and major external elements of a construction project are modelled in BIM software (Braun & Borrmann, 2019). As a result, 4D BIM and computer vision integration to update the schedule of the exterior construction environment has been advanced (Han & Golparvar-Fard, 2017; Z. Wang et al., 2021). The merits and drawbacks of computer vision based technology for progress monitoring studies are discussed in Table 2.1. In addition to computer vision technology, other state-of-the-art methods that use laser scanning, radio waves and geospatial data have improved the automation levels in the process of monitoring construction progress.

Omar and Nehdi (2016) and Reja et al. (2022) identified the crucial factors for selecting the data collection technology: the level of automation required for data capturing, equipment cost, time taken for calibration, data capturing and processing, real-time data availability, level of accuracy, data capturing range and angle, lighting requirement for data capturing, operator/user training requirement and equipment portability. Although these requirements are based on an individual project and the expectations of the progress monitoring process, the sensor mounting methods can be considered to facilitate the operation of data capturing technologies. Crucial factors for the sensor mounting methods include the cost of the mounting equipment, range of operation, the nature of the case project, operator/user training, flexibility in navigation, manoeuvring speed, accessibility

and statutory clearance (Reja et al., 2022) The sensor mounting methods for data collection are fixed devices, handheld devices, robotic systems mounted on unmanned ground vehicles (UGV), unmanned aerial vehicles (UAV), or a mixture of these systems (Omar & Nehdi, 2016; Reja et al., 2022). Considering the choice of the data acquisition technology and the sensor mounting methods, the merits and limitations are summarised in Table 2.1.

Table 2.1: Data acquisition technologies for progress monitoring

Data acquisition method	Computer vision based technologies	Laser technology	Barcoding	RFID	UWB	GPS and GIS
Merits	- No extensive training required - Comparatively cheaper - High portability and handheld ability - Can be used in both exterior and interior environments (Bohn & Teizer, 2010; Lu & Lee, 2017)	- High accuracy - Rapid 3D reconstruction ability (Golparvar-Fard, Peña-Mora et al., 2011; Lee & Park, 2019)	- Easily identified using mobile devices - High portability and handheld ability (Y.-C. Lin et al., 2014)	- Wide reading range - High portability and handheld ability - Durability in construction environments under harsh weather conditions (Omar & Nehdi, 2016)	- Ability to operate both indoors and outdoors - Low power requirement (Shahi et al., 2013)	- Effective in capturing location and mapping data - Covers a wide range (Ko, 2013)
Limitations	- Fixed cameras have limited coverage - Continuous relocation of cameras is labour intensive (Bohn & Teizer, 2010; Dai et al., 2012; Teizer, 2015)	- High cost - Needs expertise to operate - Requires a clear line of sight - Noise in raw point clouds reduces accuracy - Requires regular sensor calibrations - Requires many data points (Bosché, 2010; Golparvar-Fard et al., 2015; Lee & Park, 2019)	- Requires a direct line of sight between the barcode tag and the reader - Limited amount of information can be stored - Information cannot be modified once it is stored - Barcode tags can easily be damaged or lost (El-Omari & Moselhi, 2011; Shahi et al., 2013)	- Relatively high initial and maintenance costs - Requires periodic battery replacement - Only captures position data - When used in indoor sites, RFID tags have a reduced communication range (Ergen et al., 2007; Navon & Sacks, 2007)	- Signals can get blocked when used indoors - Only captures position data - Registration is challenging and requires complex algorithm (Ham et al., 2016; Kropp et al., 2018)	- In congested environments, positioning can be challenging due to the distortion of satellite signals - Usually ineffective to operate indoors - System becomes expensive when a GPS receiver needs to be attached to each site object (Kang & Tesar, 2004; Khoury et al., 2015)

As summarised in Table 2.1, computer vision based data capturing methods such as digital cameras offer competitive advantages compared to other data capturing methods with the right mounting method, particularly in indoor construction environments. Collecting meaningful as-built data by using state-of-the-art technologies has a great impact on analysing the as-built status of construction sites. With the advances in computer vision based image processing algorithms, recognition of construction objects from image data for as-built status analysis has been achieved with or without a prior model such as a BIM (C. Liu et al., 2021; Martinez et al., 2021; Roberts & Golparvar-Fard, 2019).

2.3 Algorithmic approaches to analyse the as-built status from image data

After capturing as-built data, recognising construction elements is critical to document the as-built status and then analyse the as-built conditions (Hamledari, McCabe, & Davari, 2017). In current computer vision based construction progress monitoring, the construction elements recognition encompasses detecting and classifying objects from as-built images using machine learning algorithms (Kardovskyi & Moon, 2021). By adding artificial intelligence approaches for image processing, automated visual analysis is performed for as-built construction elements (Hamledari & McCabe, 2016; Hamledari, McCabe, & Davari, 2017).

2.3.1 *Conventional machine learning approaches*

When administering detection and classification of objects in images, conventional machine learning algorithms use manual feature extraction, which is also referred to as handcrafted feature extraction. The manual feature extraction entails the programmer designing the specific features to be extracted from the images (O'Mahony et al., 2019). Features are small descriptive informative patches such as edges, lines, colours, and texture (Nanni et al., 2017). For example, the feature extraction algorithm Canny edge detector (Canny, 1986) extracts edges or lines that can be used to classify and detect objects in which the edges are the prominent feature (Nanni et al., 2017). To recognise framing instance in construction sites, studs have been detected by the edges using Canny edge detector as presented in Figure 2.8.

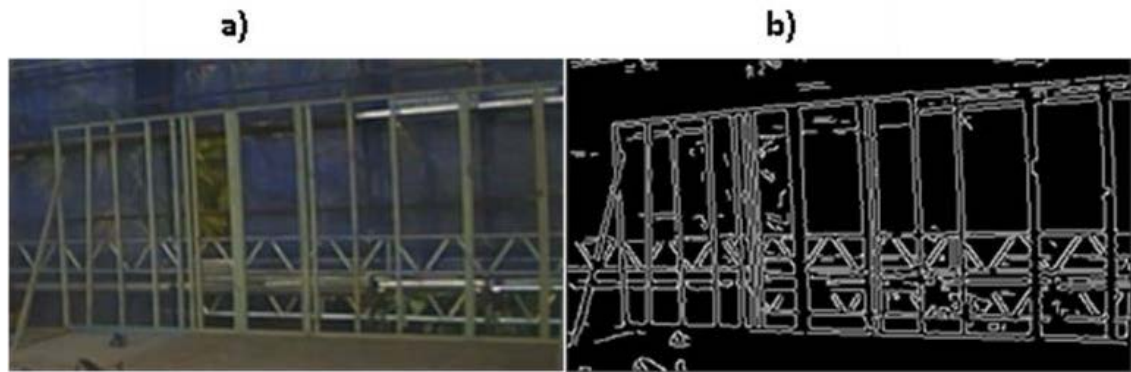


Figure 2.8: Feature extraction: a) Original image; b) Subjected to Canny edge detection

Source: (The Author)

The conventional machine learning approach requires manually selecting the features, which are important in each image if the images have different levels of exposure, occlusions, and illumination (M.-Y. Chen et al., 2020). When the number of object classes increases, feature extraction becomes cumbersome (O'Mahony et al., 2019). Although the object detection and classification are effective in automatically differentiating the construction element's as-built states from site images, calculation of work completion of as-built elements requires the shape extraction of the objects. This can be achieved by using a pixel level masks approach of image segmentation (K. He et al., 2017).

Compared to object detection and classification, segmentation is a complicated task for traditional machine learning algorithms (Bhargavi & Jyothi, 2014). Image segmentation is the computer vision solution to detect and classify the individual objects in an image at the pixel level (K. He et al., 2017). To achieve traditional algorithmic based segmentation, the original image must be converted to a binary image (Munawar et al., 2021). This process is manually done by the programmer (Bhargavi & Jyothi, 2014; Iannizzotto & Vita, 2000). Among the conventional algorithmic approaches for image segmentation, thresholding using the Otsu method, and edge based, and region based segmentation are prominent (Xi et al., 2020). Otsu based thresholding is the simplest and the fundamental form of segmentation to split an image into segments (Xu et al., 2011). From a grey scale image of the original image, thresholding is used to generate binary images to define the boundaries of the segments (Bhargavi & Jyothi, 2014). In edge based segmentation, the input image is converted to a binary image and the points are obtained through contour marking using edge extraction (Iannizzotto & Vita, 2000). For region based segmentation, pixels corresponding to an object in the image are clustered together and marked to

achieve segmentation (Kaganami & Beiji, 2009). A binary image is generated from the original image in order to segment the insulation area as depicted in Figure 2.9.

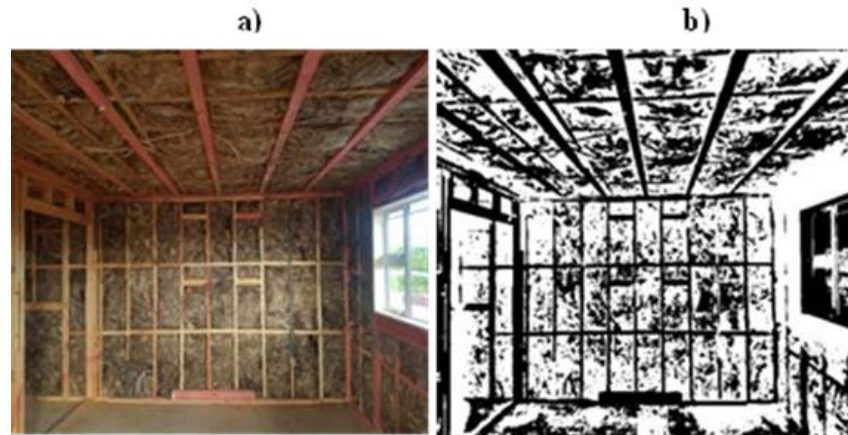


Figure 2.9: Image segmentation: a) Original image; b) Binary image

Source: (The Author)

When processing images for detection, classification and segmentation using traditional machine learning algorithms, in addition to the information on shape, colours, texture, location and orientation of objects, a significant level of human pre-processing is required to remove background noise and to enhance visual quality (Razavi et al., 2008). Generally, pre-processing related to lighting variation is done using low-light image enhancement (LIME) methods. This is used to enhance the images captured in low-light conditions such as night-time (Guo et al., 2016). Gamma correction is generally applied to reduce the light effect by altering the pixel value (Xu et al., 2009). To remove noise resulting from cluttered scenes and background objects, techniques such as kernel density estimation are used (Q. Wang et al., 2019). To separate moving foreground objects from the non-moving background in video processing, background subtraction techniques such as frame differencing is employed (Kartika & Mohamed, 2011). This pre-processing is performed to provide the programmer with clear and occlusion free features of objects in the images (Razavi et al., 2008). Despite the advances of machine learning, traditional algorithms have limitations in extracting features manually, and become ineffective when an image has multiple instances from different object classes and overlapping objects to be segmented (Xi et al., 2020). In recent years, the deep learning approach, a branch of machine learning, has been further developed and increasingly applied to construction progress monitoring tasks (Martinez et al., 2019).

2.3.2 Deep learning approaches

Deep learning models, which are also referred to as deep neural networks due to their neural network layers architecture, leverage input-to-target mapping through a deep sequence of layers to extract features from input data (Chollet, 2017). Object recognition using deep learning aims to locate, classify, and detect objects in the images and label them with rectangular bounding boxes with their confidence score of existence. As per supervised learning, to train a deep learning model, a large amount of data must be gathered and labelled (Chollet, 2017; LeCun et al., 2015). This mechanism is different from the manual feature extraction in conventional machine learning models, in which the features are selected by the programmer. The difference in the mechanism behind conventional machine learning and deep learning models in recognising objects is illustrated using a framing example in Figure 2.10.

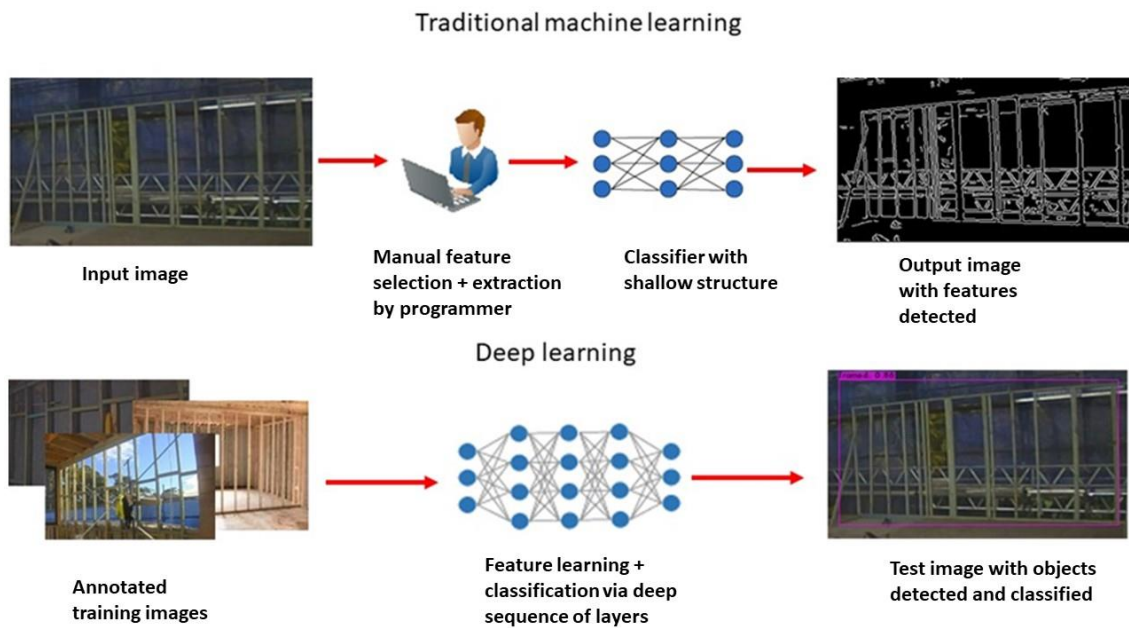


Figure 2.10: Mechanisms: a) Traditional machine learning; b) Deep learning

Source: (The Author)

Since the framing studs encompass edges as the main feature, an edge detection algorithm can be used to select the edges in the studs manually by the programmer. Such algorithms use a shallow classification structure to determine the object class (Chen et al., 2020). When using deep learning to detect and classify the framing instance, under supervised learning, a large, annotated framing image dataset is provided for the deep learning model

to train on. The deep learning model automatically learns the features and classifies the object (L. Liu et al., 2020).

In order to apply the deep learning models in various contexts, understanding the mechanism behind deep neural networks is important. As explained by Chollet (2017) and Goodfellow et al. (2016), the mechanism behind deep learning neural networks is further discussed. This mechanism is illustrated in Figure 2.11. The layers of the deep neural network are parameterised by the weights of random values. The learning in deep learning refers to determining the values for the weights of the layers, so the neural network will accurately map inputs to their related targets. Initially, random values are assigned to the weights and the neural network implements a series of random transformations. To control the output of a neural network, it is important to measure how far the prediction of the neural network is from the intended output. The loss function measures the predictions and the targeted outputs. Then a distance score is computed to capture the performance of the network. Initially, the predictions are far from the intended because the layers have originally been assigned with random weight values. As a result, the loss score is high. Then this loss score is used as a feedback signal to modify the value of the weights such that the loss score is minimised. This adjustment is an optimiser function which implements the backpropagation. The amount that the weights are updated during training is known as the learning rate.

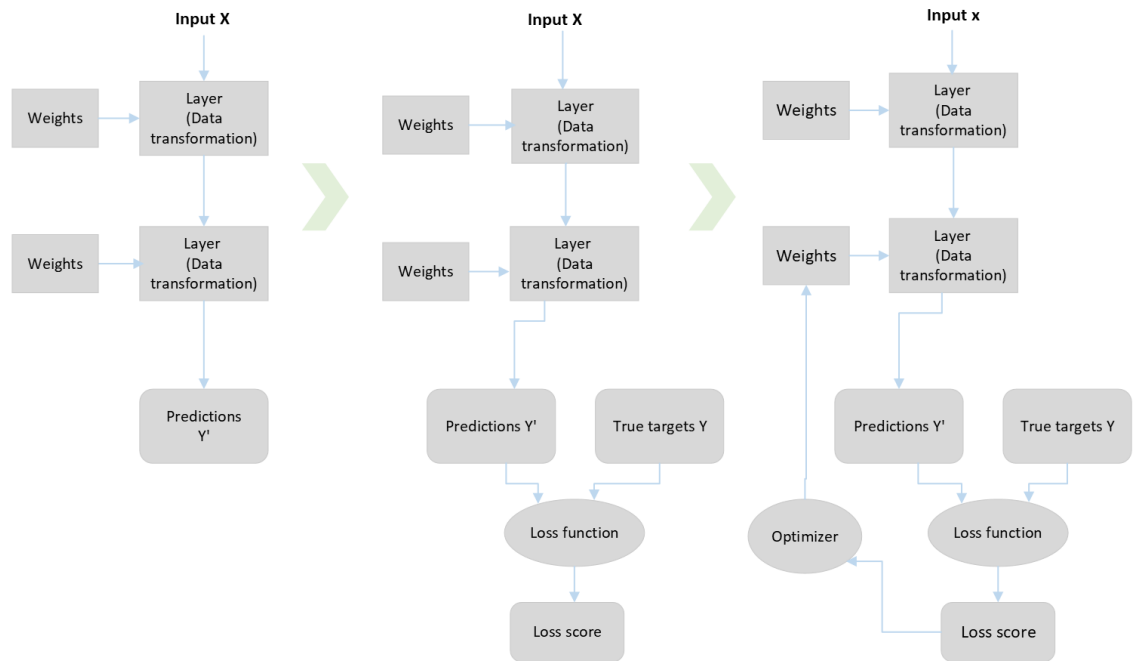


Figure 2.11: Mechanism behind deep neural networks

Source: Chollet (2017)

With each input the neural network processes, the weights are adjusted, and the loss score is reduced. This is known as the training loop in which several iterations are repeated to yield weight values that minimise the loss function. A trained neural network has a minimal loss and produces outputs that are close to the intended targets. This ability to train the deep neural network to learn the features of objects has eliminated the manual interventions of feature extraction (Praveen Gujjar et al., 2021).

2.3.3 Convolutional neural networks

Convolutional neural networks (CNNs) are a widespread type of deep learning neural network used for image processing (Fang et al., 2018; J. Kim et al., 2018). CNNs achieve better object recognition accuracy on large datasets with the joint feature and classifier learning ability. CNNs are capable of accurately and efficiently localising objects in images (Gu et al., 2018). A CNN adopts a cascade of multiple layers, which encompass convolution calculations of processing units for feature extraction from images. A typical CNN structure is composed of an input layer, convolutional layers, pooling layers, fully connected layers, and an output layer. Between the input layer and output layer, the other cascaded layers are hidden layers (Chollet, 2017; Gu et al., 2018; J. Han et al., 2018; LeCun et al., 2015; J. Liu et al., 2021).

The typical structure of a CNN, the hyperparameters and the functions of the hidden layers as explained by Chollet (2017), J. Liu et al. (2021) and LeCun et al. (2015) are discussed and further illustrated in Figure 2.12. The input layer of a CNN represents the pixel matrix of the input image. Then the convolutional layer extracts features by using convolution kernels and outputs a feature map. The feature map is created by first convolving the input with a learned kernel and then applying an element-wise nonlinear activation function on the convolved results. The activation function defines how the weighted sum of the input is transformed into an output from a node in a layer. Typical activation functions are sigmoid, tanh and rectified linear unit (ReLU) (Sibi et al., 2013). A pooling layer is generally placed between two convolutional layers. The conventional pooling operations are average pooling and max pooling. The pooling layer reduces the length and width of the previous feature map and enhances the efficiency of the calculation by down-sampling. The kernels in the first convolutional layer are devised to detect low-level features such as edges and curves, while the kernels in higher layers are designed to learn more abstract features. After several convolutional and pooling layers, there may be one or more fully connected layers to perform high-level reasoning. The last layer of a CNN is an output layer, which uses a logistic function or a softmax function to output classification labels. The probabilities for all output nodes are also calculated in this process. The feature learning and extraction layers of the CNN are generally referred to as the network backbone. The layers responsible for predicting classes and bounding boxes of the objects are called the network head (Bochkovskiy et al., 2020).

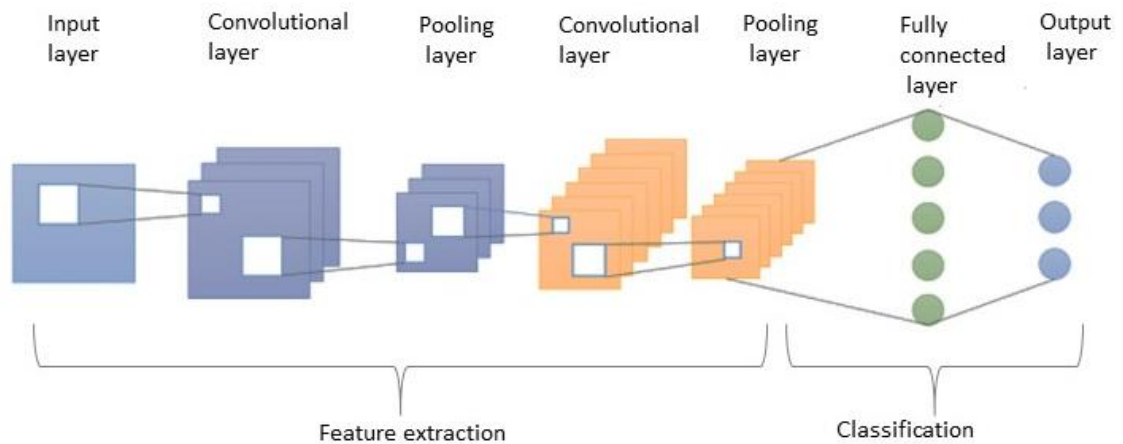


Figure 2.12: Typical structure of a CNN and the functions of the hidden layers

Adapted from: Chollet (2017); LeCun et al. (2015).

Optimisation is critical for minimising the loss function. The stochastic gradient descent (SGD) is the traditional and common optimiser used for many CNNs (LeCun et al., 2015). The Adam optimiser (Kingma & Ba, 2014) is an extension to SGD that has recently been adopted in deep learning models to update the network weights iteratively. Batch normalisation is used to stabilise the learning process and reduce the number of training epochs required to train (Gu et al., 2018). To produce high performing and accurate CNNs, in addition to the deep sequence of layers, large datasets for training are also essential (J. Han et al., 2018).

In CNN, there are two types of frameworks. In the first type of CNN framework, object detection is regarded as a regression or classification problem. You Only Look Once (YOLO) (Redmon et al., 2016), and single shot multi-box detector (SSD) (W. Liu et al., 2016) are two examples. In the second type, region proposals are generated, and each proposal is classed into different object categories. Region proposals are generated to predict in which regions of the image the objects may exist (Kardovskyi & Moon, 2021). Region based convolutional neural networks (R-CNNs) such as Fast R-CNN, Faster R-CNN, and Mask R-CNN are some examples (Zhao et al., 2019).

The application of deep learning based CNN models for construction progress monitoring has gained momentum in recent years. Examples include rebar counting using YOLO in Y. Li et al. (2021), pre-cast wall installation monitoring using Mask R-CNN in Z. Wang et al. (2021), floor panel installations using a R-CNN based model in Martinez et al. (2021) and reinforcement, formwork and concreting works as-built progress calculations using Mask R-CNN in Pal et al. (2022). With the development of algorithmic approaches to analyse the as-built state from traditional machine learning algorithms to deep learning algorithms, computer vision based progress monitoring studies have been implemented at an escalated pace. However, the majority of computer vision based progress monitoring studies have focused on the exterior construction environment and few studies have been conducted for indoor sites (Deng et al., 2020; Hamledari & McCabe, 2016; Kropp et al., 2018).

2.4 Indoor construction progress monitoring

The scope of indoor construction progress monitoring generally covers the interior of building projects, fit-out projects, refurbishment, renovation and retrofitting projects (Hamledari et al., 2018; Kropp et al., 2018). According to Rawlinsons (2015, p.75), a

commercial high-rise building in Australia has a percentage cost per square metre share of approximately 40% for building finishes and services. Even though some solutions that are used in exterior construction environments can also be applied in interior environments, they do not fully resolve the challenges encountered in interior environments (Kopsida et al., 2015). This is because of the significant physical differences between the exterior and interior construction environments (Hamledari & McCabe, 2016; Hamledari, McCabe, & Davari, 2017). On the other hand, construction is often presumed to be an outdoor activity (McCabe et al., 2017). As a result, computer vision based indoor construction progress monitoring tends to be undervalued (Hamledari & McCabe, 2016; Kropp et al., 2018). Even highly reliable computer vision based systems may succumb to inefficient visual recognition of the as-built elements in dynamic environments such as indoors (Wong et al., 2022). A few studies on computer vision based indoor construction progress monitoring have been conducted between 2009 to 2022. computer vision based indoor construction progress monitoring studies are discussed in the three subsections below. In the first part, the advances in indoor image collection and visualisation are reviewed. In the second part, studies related to automated activity state recognition of indoor works are presented. In terms of algorithmic processing, all these studies have used handcrafted feature extraction methods. The adoption of deep learning for indoor construction progress monitoring is reviewed in the third subsection. An overview of these studies is presented in Table 2.2.

Table 2.2: Overview of computer vision based indoor construction monitoring

Key emphasis	Data capturing method	Key algorithms	Source
Unmanned vehicle based image capture platforms			
Potential of UAV for indoor site monitoring	UAV mounted camera	-	(McCabe et al., 2017)
Accuracy of partitions progress tracking with respect to the UAV's velocity and photo capture configuration	UAV mounted camera	-	(Hamledari, McCabe, Davari, Shahi, et al., 2017)
Registering video frames to a BIM to detect and match the vanishing points and vanishing lines of the image frames	Videos captured from an UGV mounted monocular camera	SLAM	(Asadi & Han, 2018)
Using UAVs and UGVs collaboratively and simultaneously for data capture	UAV is mounted with a wide-angle camera and UGV is mounted with a LiDAR	-	(Asadi & Han, 2020)
Visual 3D mapping of indoor locations and image localising			

Comparing algorithms for proper estimation of camera localisation	Handheld camera	SFM, VO and VSLAM	(Kropp et al., 2016)
Leveraging the mapping strengths of SFM and MVS and real-time localisation strengths of SLAM to monitor scattered work locations	Handheld camera	SFM, MVS, SLAM, Direct Linear Transformation and Single Value Decomposition, ORB and HaHoG features	(Amer & Golparvar-Fard, 2018a)
Developing a hybrid SFM-SLAM 3D reconstruction algorithm	Handheld camera	Hybrid SFM-SLAM 3D reconstruction algorithm built on ORB features, MVS, RANSAC	(Amer & Golparvar-Fard, 2018b)
Determining the camera poses of image frames in BIM coordinate system by performing an augmented monocular SLAM	Handheld camera	Augmented monocular SLAM, multi-view environment, Canny edge detector, voting and searching schemes, gradient descent based algorithm, perspective projection using vanishing points and lines	(Asadi et al., 2019)
Visualisation of progress using 3D walk-through models			
Prototype of 3D walk-through model to visualise progress discrepancies	Digital camera with a wireless transmitter	-	(Roh et al., 2009)
3D walk-through modelling with AR and BIM for activity monitoring, where activities are treated as objects	Digital camera with a wireless transmitter	Cascade of Haar like feature-based classifiers	(Roh et al., 2011)
Automated activity state recognition of indoor works			
Conceptual framework to predict finishing delays based on item's presence and state and generate an as-built 4D BIM	Handheld camera	-	(Kropp et al., 2012)
Pre-processing images with filters to reduce object detection to a classification problem and estimating correspondences between images and 4D BIM	Handheld camera	Perspective-n-point algorithm, HoG features, SVM classifier	(Kropp et al., 2013)
Determining the states of drywall completion using line and dot characteristics and reflecting properties of materials	Handheld camera	Compass edge distribution and pixel intensity histogram features, CLAHE parameters, SVM classifier	(Kropp et al., 2014)

Detecting steel studs and electrical boxes using an integrated shape and colour-based approach	UAV mounted camera	CIELAB, Otsu thresholding, morphological transformations, Hough Transform, Canny edge detector	(Hamledari & McCabe, 2016)
Automatically detecting partition's current state using four computer vision based modules based on an algorithm which uses edges, texture and colour features of each module	UAV mounted camera, smartphone	HoG, bi-lateral filter, Otsu cluster-based image thresholding, L-channel, PPHT, voting system, k-means clustering, Gaussian filter, CIELAB, LBP, SVM classifier, CLAHE, Sobel kernel, multistage thresholding algorithm, CDF histogram, inverted binary thresholding, morphological transformations	(Hamledari, McCabe, & Davari, 2017)
The object's progress ratios are automatically incorporated in BIM	UAV mounted camera	HoG, bi-lateral filter, Otsu cluster-based image thresholding, L-channel, PPHT, voting system, k-means clustering, Gaussian filter, CIELAB, LBP, SVM classifier, CLAHE, Sobel kernel, multistage thresholding algorithm, CDF histogram, inverted binary thresholding, morphological transformations	(Hamledari, McCabe, Davari, & Shahi, 2017)
Each image of a sequence is registered to the 4D BIM in terms of its origin and its point in time in the construction schedule	Handheld camera	Line segment extraction approaches, camera pose estimation approaches i.e., modify RANSAC for pose estimation, VO, SfM and HOG features, compass edge distribution and pixel intensity histogram features, SVM classifier	(Kropp et al., 2018)
Determining the engineering quantity in real time for tiling activity which is calculated by area	2D photos captured from fixed site cameras	SVM classifier based on LBPs, canny edge detection algorithm Hough transform algorithm, camera calibration algorithm, tile orientation determining algorithm	(Deng et al., 2020)
Developing an image enhancement approach to improve visual recognition of images, which are captured in low lit indoor environments	2D photos captured from fixed time-lapse cameras	Refined approach of LIME and Gamma correction	(Wong et al., 2022)

Deep learning for indoor construction progress monitoring

Mask R-CNN based approach to construct as-is BIM objects from indoor images of an existing building	Handheld camera, smart phone, and online images	Mask R-CNN	(Ying & Lee, 2019)
Mask R-CNN based automated recognition and segmentation of air conditioning ducts	Synthetic images from BIM models	Mask R-CNN	(Shamsollahi et al., 2021)
Mask R-CNN for calculating the work-in-progress of brick layering and plastering in an indoor wall	Zed2 camera and online images	Mask R-CNN	(W. Wei et al., 2022)

Legend:

CDF – Cumulative distribution function
 CLAHE – Contrast-limited adaptive histogram equalisation
 HOG – Histogram of oriented gradients
 IMU – Initial measurement unit
 LBP – Local binary patterns
 LiDAR – Light detection and ranging
 Mask R-CNN – Mask regional convolutional neural network
 MVS – Multi view stereo
 ORB – Oriented FAST and Rotated BRIEF
 PPHT – Progressive probabilistic Hough transform
 RANSAC – Random sample consensus
 SLAM – Simultaneous localisation and mapping
 SFM – Structure from motion
 SVM – Support vector machine
 UAS – Unmanned aerial system
 UAV – Unmanned aerial vehicle
 UGV – Unmanned ground vehicle
 VO – Visual odometry
 VSLAM – Visual simultaneous localisation and mapping

2.4.1 Advances in indoor image collection and visualisation

To facilitate computer vision based indoor construction progress monitoring, advances include the use of cameras, different camera positioning platforms for data collection, estimation of camera poses through localisation, and visualisation of progress by integrating with BIM. These advances in computer vision based indoor construction progress monitoring studies are reviewed in this section.

The use of fixed cameras in indoor sites can be challenging due to the limitations in range and continuous relocation (McCabe et al., 2017). These limitations can be remedied by using image capturing platforms such as unmanned aerial vehicles (UAV) and unmanned ground vehicles (UGV) (Asadi & Han, 2020; Ibrahim et al., 2019). Although unmanned vehicle based data capturing platforms have distinctive advantages over fixed cameras, the disadvantages stemming from their measuring positions in the indoor environment (P. Kim et al., 2016) should also be taken into account. A ground robot or vehicle can be

more easily stabilised with higher payloads than an aerial robot or vehicle (Ibrahim et al., 2019; P. Kim et al., 2019). The UAVs typically use a camera to capture image data. Moreover, UGVs have a stop in-place safe strategy enabling dynamic obstacle avoidance unlike air collisions caused by UAVs in a cluttered indoor environment (Ibrahim et al., 2019). However, the ground based measurement methods used by UGVs are affected by blind spots behind vertical obstacles and they are unable to capture horizontal surface data located higher than the UGV. In contrast, UAVs can even capture horizontal obstacles such as a roof (Ibrahim et al., 2019).

Localising the camera poses with respect to a BIM in real time is challenging in cluttered indoor environments (Asadi & Han, 2018). Because of their in-flight agility and capacity to hold cameras while flying, UAVs are efficient for data capturing (Hamledari, McCabe, Davari, Shahi et al., 2017; Hamledari et al., 2018; McCabe et al., 2017). The UAV's velocity and photo capture configuration are crucial for the accuracy of automatically detecting the components of indoor partitions such as studs, insulation, electrical outlets, and the state of drywalls (Hamledari, McCabe, Davari, Shahi et al., 2017). However, UAVs can pose safety risks to construction workers due to their flight path and can cause distractions with their noise. On the other hand, although the layout of the project is initially known, it can change affecting the flight path. Moreover, UAVs' rotational velocity and sudden angular movements can cause motion blur in the captured images (Hamledari, McCabe, Davari, Shahi et al., 2017; Hamledari et al., 2018; McCabe et al., 2017). Both UAVs and UGVs need expert operators and they are relatively expensive compared to ubiquitous site surveillance cameras (Asadi & Han, 2020).

Indoor construction activities are mostly performed by different teams working in dispersed locations. A large number of images is needed to map all locations, particularly corridors and hallways (Amer & Golparvar-Fard, 2018b). To properly estimate the camera localisation within indoor sites, computer vision techniques such as structure from motion (SFM), visual odometry (VO) and visual simultaneous localisation and mapping (VSLAM) have been tested using the images captured from handheld cameras (Kropp et al., 2016). SFM reconstructs the camera motion and reveals the observed structure of the scene, whereas VO estimates the path of the camera that provides an image sequence. VSLAM estimates the path of the camera while creating a map (Kropp et al., 2016).

To increase the efficiency of image localisation, a hybrid SFM-SLAM 3D reconstruction algorithm (Amer & Golparvar-Fard, 2018a) has been proposed to map the work done at two different and scattered indoor locations and registering them together in BIM (Amer & Golparvar-Fard, 2018a). After generating a locally dense as-built map of all work locations, such as corridors, hallways, and other connecting areas, registering to BIM allows for as-planned and as-built analysis (Amer & Golparvar-Fard, 2018b). As an extension to previous works, an augmented VSLAM was introduced to improve camera trajectory and to discover camera poses of image frames in the BIM coordinate system (Asadi et al., 2019).

Among the initial attempts to enable 3D walk-through visualisations in indoor environments, the prototype introduced by Roh et al. (2009) is prominent. The prototype uses interior construction images superimposed over a 3D model and user's viewpoint information to visualise the indoor construction site in an augmented reality (AR) 3D walk-through model (Roh et al., 2009). This prototype has been improved to represent the progress of an activity completion, in which the activities are treated as objects (Roh et al., 2011). As-built construction photographs of air conditioning ducts captured using a handheld camera with a wireless transmitter are superimposed over an as-planned BIM to visualise progress states in terms of colour codes. Due to the object based approach in representing activities, the ability to detect deviations of processes and objects that are not modelled in BIM is limited. The approach also requires the user to manually collect image data and enter spatial-temporal information of time, location, and viewpoint for each photograph, making this approach manually intensive (Roh et al., 2011).

Advances in data collection and visualisation of progress information in indoor construction sites have contributed to improve activity state recognition of indoor works through the adoption of traditional machine learning algorithms for indoor object detection and classification.

2.4.2 Automated activity state recognition of indoor works

The use of computer vision based real-time monitoring has been explored for the as-built progress recognition of indoor works (Deng et al., 2020; Hamledari & McCabe, 2016; Hamledari, McCabe, & Davari, 2017; Kropp et al., 2012; Kropp et al., 2014; Kropp et al., 2018). These past studies have exploited distinctive features related to characteristics such

as shape, colour, and texture of indoor elements (Hamledari & McCabe, 2016), thus employing handcrafted features to achieve activity state recognition (Nanni et al., 2017).

With the aim of automatically predicting delays in finishing works, a conceptual framework using video data and 4D BIM has been proposed focusing on installation of heaters, electric outlets, painting, and tiling (Kropp et al., 2012). This approach proposes to estimate the handheld camera motion with reference to BIM, as the user traverses the indoor site (Kropp et al., 2012). Further to that, an approach for activity state recognition of a heating device, which uses the data from schedule loaded BIM models and motion registered videos, has been presented (Kropp et al., 2013). This approach discusses how the image data is pre-processed with filters, so that the images are prepared for robust object detection. The histogram of oriented gradients (HOG) features are extracted to train and classify image data using support vector machine (SVM), reducing object detection to a simple 2D classification problem (Kropp et al., 2013). These studies have been further improved to determine the different states of drywall completion using a two-stage cascaded approach (Kropp et al., 2014). The camera position and the pose are required to be known from the motion estimation. A 4D BIM model is used to derive drywall extents and identify the region of interest. The first stage focuses on the line and dot characteristics to identify the edges for detecting panelled drywalls. The second stage determines the difference of the reflecting properties of materials on plastered drywalls. Using pixel intensity, plastering, and painting is distinguished (Kropp et al., 2014).

The study of Kropp et al. (2018) integrates the previous studies of Kropp et al. (2012), Kropp et al. (2013) and Kropp et al. (2014) for robust radiator state recognition (unprepared background, prepared background, radiator wrapped in packaging foil, and unwrapped radiator) and drywall completion (installed drywall panels, plastered drywall and painted drywall). This approach enables registering each image of a sequence to the 4D BIM in terms of its origin and its point in time in the construction schedule. Taking the as-built video as input, the first step of this approach registers image data to the BIM. After registration, the recognition step represents the actual state of activities (Kropp et al., 2018).

When detecting components of interior partitions such as steel studs and electrical boxes through the images captured using an UAV mounted camera, the distinctive features of each component have been exploited using a shape and colour based approach (Hamledari

& McCabe, 2016). The object detection procedure first considers differentiating the objects from the background using colour and illumination intensities. Second, it carries out processing of the extracted shape and localises the objects of interest using Canny edge detector and Probabilistic Hough Transform. The edges, shape, texture, and colour based feature extraction has been employed to monitor the as-built states of indoor partition elements such as steel studs, batt insulation, electrical outlets, and three states for drywall sheets (Hamledari, McCabe, & Davari, 2017). The algorithm detects the studs, then the insulation followed by the drywall state completion and electrical outlets installation using feature extraction. The element's progress ratios are automatically incorporated to 4D BIM. Colour coding is used to reflect progress deviations (Hamledari, McCabe, Davari, & Shahi, 2017).

Extending the application of indoor object detection to work-in-progress calculations, Deng et al. (2020) introduced support vector machine (SVM) classifier-based local binary patterns (LBP) to classify tile and non-tile areas captured through images from a fixed camera. The contours of tiles are identified through an edge detection algorithm and the straight lines are separated from these contours by a Hough Transform algorithm. In the as-planned BIM model, the location of the paved tiles is known, thus the boundary line of the tile paving area identified through on site images should correspond to the tile segmentation line in the BIM (Deng et al., 2020). Among the other recent studies, to facilitate recognition of indoor objects in low illuminated environments, Wong et al. (2022) introduced an algorithm named enhanced illumination quality based on the improvements of LIME and Gamma correction. The algorithm has been tested on raised floor panels in an indoor site.

2.4.3 Adoption of deep learning in indoor construction progress monitoring

The deep learning algorithms, which are capable of image segmentation can transform the existing construction progress reporting of completed or non-completed states (Pal et al., 2022). The recent computer vision based indoor construction progress monitoring has harnessed the detection and classification ability of Mask R-CNN to recognise indoor elements with irregular shapes (Shamsollahi et al., 2021; Ying & Lee, 2019). The segmentation ability of Mask R-CNN has been used to calculate partial work completion percentages (W. Wei et al., 2022).

Although BIM models can facilitate as-built and as-planned progress comparisons, some construction stages are usually not modelled in BIM (Pal et al., 2022). This is generally applicable for indoor construction works (W. Wei et al., 2022; Ying & Lee, 2019). Therefore, a Mask R-CNN based approach has been developed to construct as-is BIM objects from indoor images of an existing building (Ying & Lee, 2019). Due to the automatic feature learning ability, Mask R-CNN based building object recognition is robust to complex environmental conditions, and scalable to customised building object types in the indoor environment (Ying & Lee, 2019). Three types of building objects (walls, doors, and lifts) have been used in this study. The approach has been conducted by recognising and segmenting the three indoor building elements in images collected from four buildings of a university. Handheld cameras, smart phones and online images have been used to create the training dataset of 1,076 images. After extracting the shape of the elements, coordinate transformation has been performed to construct the object's geometry and to generate the BIM objects corresponding to walls, doors, and lifts. The mean average precision (mAP), which is used to measure the accuracy of the deep learning model, is 91.2% (Ying & Lee, 2019). Despite the pioneering study of applying deep learning models for indoor elements recognition, this study only provides evidence on the as-is status of the indoor elements, while the progression of the as-built construction and the work-in-progress percentages are not reported. The automation level requires heavy manual intervention during the data collection using handheld cameras and smart phones.

In another study to recognise indoor objects with irregular shapes, Mask R-CNN has been used for automated recognition of air conditioning ducts. 3D BIM models have been used for generating 1,143 synthetic images to train the Mask R-CNN model (Shamsollahi et al., 2021). For an image dataset with augmented images, the trained Mask R-CNN model reports a mAP of 90.6%. Despite the contribution of this study in automated recognition of air conditioning ducts, which is a complex indoor element, since synthetic images are used for the training image dataset, this study fails to reflect the impacts of challenging indoor construction environments. Moreover, this study fails to provide evidence on the progress of the as-built construction of indoor elements.

To harness the segmentation ability of Mask R-CNN for calculating the work-in-progress, indoor wall construction through brick laying and plastering is the focus in W. Wei et al. (2022). The progress data have been captured using a handheld stereo camera (Zed2),

which has the ability of position tracking. A Mask R-CNN model has been optimised through fine-tuning three hyperparameters of loss function improvement, SGD optimiser improvement, and cosine annealing. This model has been trained on 500 images. These images have been captured from a building site and sourced from the internet. The Mask R-CNN model has been tested on the images from the same building site. As a result, the model reports a mAP of 96.8% (W. Wei et al., 2022). To determine the percentage work-in-progress of an indoor wall, the pixel segmentation based mask area calculation has been used. Due to the position tracking ability of the camera, 2D images can be streamlined to 3D BIM through an algorithmic process proposed by W. Wei et al. (2022). This study has facilitated the visualisation of the as-built status through colour codes assigned to completion percentages. However, the automated pixel level area calculations of the segmented masks have not been demonstrated. W. Wei et al. (2022) implemented two types of algorithmic pre-processing in their study. Perspective transformation has been used to eliminate image distortion and angle effects due to camera positioning. This step is a mandatory manual intervention since the as-built wall components are mapped with Revit components. Brightness equalisation has also been used as a pre-processing technique to reduce the impacts due to brightness changes in the images. Since brightness exposure of one image is different from another image, this pre-processing had manual intervention. The manual intervention in the data collection and algorithmic pre-processing steps, the use of a small training dataset, and the testing and training images being collected from the same location are some of the drawbacks of this study. Nevertheless, W. Wei et al. (2022) offer an efficient algorithm for mapping the relationship between 2D indoor images and a 3D BIM.

Although the competitive advantages of applying deep learning models for indoor construction progress monitoring have been highlighted in the studies of Shamsollahi et al. (2021), W. Wei et al. (2022), and Ying and Lee (2019), these studies also point out the configuration difficulties in building deep learning models and the complexities in training them. Various computer vision based applications for indoor construction progress monitoring have been progressing over the past decade. The advances in these studies from applying traditional machine learning algorithms to deep learning algorithms

in combination with different camera mounting techniques are summarised with the corresponding years of studies to form a timeline in Figure 2.13.

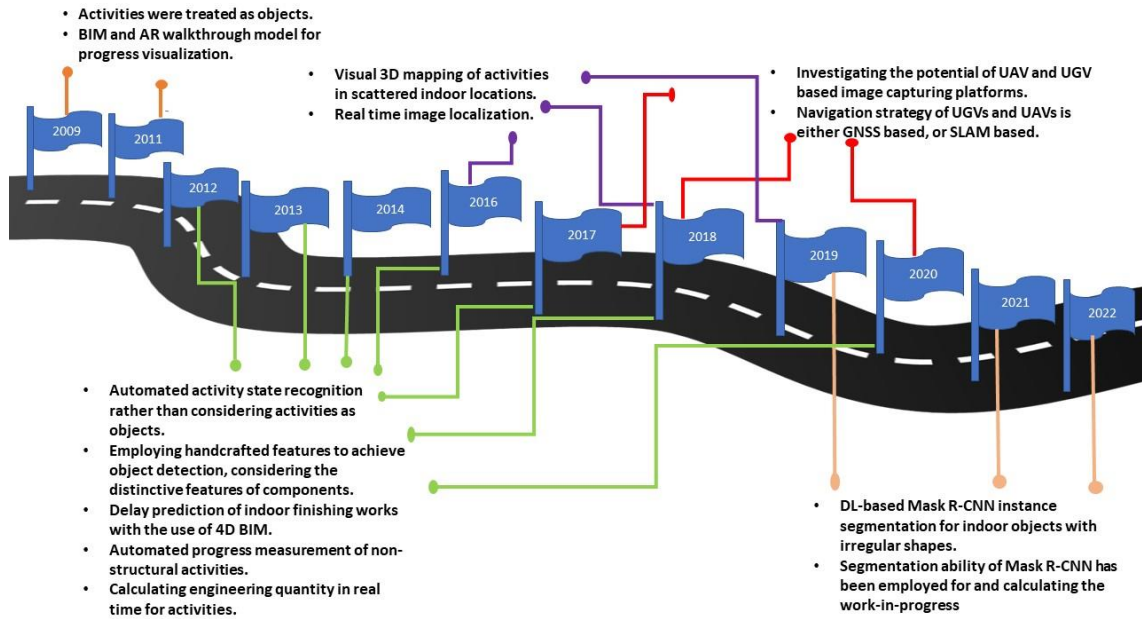


Figure 2.13: Timeline of the computer vision based indoor progress monitoring studies

Source: (The Author)

Despite the contribution of these past studies, it can be noted that the use of handcrafted feature extraction through machine learning algorithms requires manual intervention, is biased, and is less accurate. The challenges related to data collection from the as-built scenes and algorithmic processing to extract information are inevitable in such computer vision based indoor construction progress monitoring studies. These challenges are referred to as technical challenges because solutions using algorithmic approaches are required.

2.5 Research gaps

From the deficiencies identified in the existing computer vision based indoor construction progress monitoring studies, three research gaps have been determined. These gaps are extensively reviewed in the following sections.

2.5.1 Technical challenges for as-built indoor elements visual recognition

Identifying the technical challenges for as-built indoor elements visual recognition is crucial. Thus, it has been identified as a research gap. The indoor construction environment is challenging for both computer vision based and non computer vision

based data capturing devices (Deng et al., 2020). Unlike exterior construction sites, the indoor construction environment poses challenges for capturing position data (Shahi et al., 2013). The global navigation satellite system (GNSS) based platforms usually fail in navigating indoors because of the blockage of satellite signals (Ibrahim et al., 2019). The concrete and steel enveloped indoor environments interrupt GPS readings. The UWB devices are similarly affected as signals are blocked by building materials (Ham et al., 2016). RFIDs can determine the presence of objects through walls and can be deployed indoors but are limited in measuring volumetric progress to efficiently indicate the progress of partially completed tasks (Kropp et al., 2018). Moreover, they require constant installation, scanning and maintenance (Shahi et al., 2015; Shahi et al., 2013).

Although laser scanners are accurate and efficient in collecting volumetric data, they are expensive and require expert operators to set scanning configurations (Shahi et al., 2012). Laser scanners tend to generate inaccurate point clouds with complications for reflective indoor surfaces such as metal studs and pipes (Golparvar-Fard, Bohn et al., 2011; Turkan et al., 2012). The registration followed by post-processing of scans is also time consuming (Ibrahim et al., 2019). The incomplete point clouds due to obscured indoor building elements (Pučko et al., 2018) and limitations related to detecting transparent indoor elements like glazed doors and windows, which do not reflect laser beams (Pu & Vosselman, 2007), impede the use of laser scanning for indoor construction progress monitoring. The computer vision based progressing methods that succeed outdoors cannot be transferred and applied indoors without accommodating the special needs of the indoor construction environment (Kropp et al., 2016). The existing studies on computer vision based indoor construction progress monitoring have identified some challenges for visual recognition of as-built indoor elements.

Complexities in detection of indoor elements occur when the equipment and temporary materials in images cause occlusions (Hamledari & McCabe, 2016). Cluttered indoor scenes are also caused by the movements of construction personnel (Kropp et al., 2012). To remove unexpected materials and equipment such as ladders, painting pots or cables, which cause occlusion on the region of interest in the images, segmentation pre-processing algorithms are applied (Kropp et al., 2012; Kropp et al., 2014). Moreover, slender objects, such as steel studs, and small objects, such as electrical boxes in high clutter indoor scenes, cause limitations for robust feature extraction and object classification (Hamledari & McCabe, 2016; Hamledari, McCabe, & Davari, 2017).

Electrical boxes change their appearance before and after their sockets are installed. Classification difficulties are caused by similarly slender shaped objects such as columns and studs (Hamledari & McCabe, 2016).

The changes in interior walls of a building under construction constrain the use of computer vision based methods (Kropp et al., 2016). In indoor construction operations, the progress is mostly associated with changes in wall surfaces, and this causes the capture of a large number of overlapping images (Hamledari & McCabe, 2016). The different states of drywall sheets (installed, plastered, and painted) are very similar in terms of shape, colour, and texture (Kropp et al., 2012). The minimally textured and achromatic objects constrain the feature extraction (Kropp et al., 2016), when different objects have similarities in colour, shape, and surface roughness (Hamledari & McCabe, 2016). This type of feature extraction applicable to indoor elements such as drywalls leads to detection and classification errors. As a result, instead of applying a single object detection algorithm to detect and classify indoor objects, the handcrafted feature extraction requires conducting significant pre-processing to make the region of interest easily detectable (Ying & Lee, 2019).

Poor lighting conditions in indoor sites are mostly related to backlight, shadows, and artificial light sources (Kropp et al., 2014; Kropp et al., 2013). Natural light entering openings like windows produces heavy backlight causing different exposure and strong light gradient in site images and constrains feature extraction. To counteract such effects, illumination is normalised by contrast stretching, which requires manual intervention to a greater degree (Kropp et al., 2013). When handcrafted feature extraction is used to differentiate objects from the background using illumination intensities, non-uniform illumination patterns cause difficulty in extracting the shape and outline of construction elements (Wong et al., 2022). The visual recognition algorithms must be adjusted according to the lighting conditions (Hamledari & McCabe, 2016). Quality and consistency of lighting affect accurate detection of the edges of the objects. Detecting the region of interest can be constrained by changing lighting conditions due to higher pixel intensity along the illuminated areas of the object (Hamledari & McCabe, 2016). When the camera is directed toward a light source, it creates an extreme illumination condition and generates false detections (Hamledari, McCabe, & Davari, 2017).

Fixed cameras lose their effectiveness when construction work moves indoors and they are hindered by the limited data capturing range and viewing angle (Kropp et al., 2013). When fixed cameras are used indoors, obstruction of line of sight is expected as walls are erected (Hamledari & McCabe, 2016). Continuous relocation of cameras can be an exhausting exercise (Kropp et al., 2013). On the other hand, the data collection process is manual because it requires someone to inspect and capture images of the active areas in the site (Roh et al., 2011). In addition, uncertainties in unsupervised camera movements and different perspectives on objects (scale, rotation, transition) affect object recognition (Kropp et al., 2013). To generate complete as-built models for indoor construction sites, a large set of overlapping images is required due to the limited visibilities and field of view of cameras (Deng et al., 2020; Kropp et al., 2014). When UGVs and UAVs are used to capture images while traversing through indoors, the images are affected by the motion blur. UAVs pose safety risks to construction workers due to their flight path and cause distractions with their noise (Ibrahim et al., 2019; McCabe et al., 2017). When UAVs and UGVs traverse the narrow corridors and small rooms of indoor sites, the field view of the moving cameras is constrained and causes unstable motion (Kropp et al., 2016; Kropp et al., 2018).

Considering the nature of these challenges, they can be categorised into technical challenges related to **indoor objects**, **lighting conditions** and **camera positioning**. The challenges identified in the existing studies of computer vision based indoor construction progress monitoring are summarised in Table 2.3

Table 2.3: Technical challenges for visual recognition of as-built indoor elements

Literature findings	Sources											
	1	2	3	4	5	6	7	8	9	10	11	12
Indoor objects related challenges												
Detection of the region of interest is constrained by the presence of temporary equipment and material		×	×			×						
Movements of construction personnel create cluttered indoor scenes and occlusions in images		×				×						
Feature extraction of slender objects and small objects is challenging for shape based classification						×	×					
Indoor objects with minimum texture and achromatic colours constrain colour and texture based object classification		×			×	×						
Changes in the interior walls cause the capture of overlapping images						×						
Some indoor objects change their appearance during the process of installation resulting in false detections						×						
Some objects are similar in shape, constraining shape based object classification						×						
Handcrafted feature extraction of indoor objects requires a lot of pre-processing to remove noise in the images and make the region of interest detectable										×		
Lighting conditions related challenges												
Detecting the region of interest can be hampered by changing indoor lighting due to variations in pixel intensities		×	×			×						
Visual recognition algorithms must be adjusted according to the lighting conditions because feature extraction is used to differentiate objects from the background using illumination intensities						×						
Artificial light sources, backlights and shadows constrain feature extraction			×	×								
Counteracting the effects of non-uniform illumination which requires manual intervention			×									

Literature findings	Sources											
	1	2	3	4	5	6	7	8	9	10	11	12
Camera positioning related challenges												
Fixed cameras get hindered by the limited data capturing range and viewing angle			×									
When fixed cameras are used indoors, obstruction to line of sight occurs as walls are being erected						×						
Continuous relocation of cameras, with the construction progress is exhausting			×									
Data collection is manual, because it requires someone to inspect and capture images of the active areas in the site	×											
Unsupervised camera movements lead to uncertainties in capturing different perspectives of the objects			×									
To generate complete as-built models for indoor sites, a large set of overlapping images is required due to the limited visibilities and field of view of cameras				×								×
When UGVs and UAVs are used to capture images while traversing indoors, the images are affected by motion blur							×				×	
While traversing narrow corridors and small rooms of indoor sites, the field view of the moving camera is constrained and causes unstable motion					×				×			
UAVs pose safety risks to construction workers due to their flight path and noise distractions							×				×	
Sources: 1-(Roh et al., 2011); 2-(Kropp et al., 2012); 3-(Kropp et al., 2013); 4-(Kropp et al., 2014); 5-(Kropp et al., 2016); 6-(Hamledari & McCabe, 2016); 7-(Hamledari, McCabe, & Davari, 2017); 8- (McCabe et al., 2017); 9-(Kropp et al., 2018); 10-(Ying & Lee, 2019); 11-(Ibrahim et al., 2019); 12- (Deng et al., 2020)												

The empirical evidence obtained in the form of site images from the indoor construction case projects are used in Chapter 4 to further demonstrate the challenges for indoor as-built visual recognition using handcrafted machine learning algorithms. Evolving from machine learning algorithms, the recent applications of deep learning models for improving automation and accuracy in indoor construction progress monitoring have been gaining momentum. This is mainly because the feature extraction process of the deep learning algorithms is not as handcrafted as the conventional machine learning algorithms and is not affected by the lighting impacts and the background noise. The deep learning models are trained with thousands of images with different degrees of lighting and background noise to learn the patterns and features. However, even with the feature learning ability of deep learning models, the impacts of technical challenges on indoor as-built visual recognition can be expected to a certain degree. Therefore, determining the impacts of technical challenges has been identified as a research gap.

2.5.2 The deep learning model to overcome the challenges

The deep learning models perform relatively better than traditional machine learning algorithms, in terms of reducing the impacts of technical challenges. W. Wei et al. (2022) proved that the accuracy and the automation levels of deep learning models for indoor progress monitoring is high. However, the challenges related to computing requirements and training efficiency of deep learning models (O'Mahony et al., 2019; Z. Wang et al., 2021) should be considered when building and training deep learning models for robust visual recognition of indoor as-built elements. Therefore, the challenges related to building and training deep learning models are investigated and strategies to overcome these challenges are discussed in the following sections.

2.5.2.1 Building deep learning models

A deep learning model can either be built from scratch or a pre-trained model which uses existing networks can be optimised by using transfer learning and fine-tuning (Simonyan & Zisserman, 2014). In the first approach, when building a deep learning model from scratch, the process involves creating the convolutional, pooling and fully connected layers with their weights, activation functions and optimisations (Goodfellow et al., 2016). This process is computationally complicated and costly as it requires extensive mathematical and programming knowledge and large high quality image datasets for training (O'Mahony et al., 2019).

The second approach uses transfer learning to refine the pre-trained model to which the new data containing previously unknown classes is introduced (Pan & Yang, 2009). The deep learning model is then fine-tuned to accommodate the requirements of the programmer (J. Liu et al., 2021). Transfer learning enables applying relevant knowledge from previous learning experience with a large dataset to a new task with a small dataset (Torrey & Shavlik, 2010). The pre-trained models include GoogleNet, AlexNet, residual network (ResNet) and Darknet (Simonyan & Zisserman, 2014) which have been trained on large datasets such as ImageNet (Deng et al., 2009) with 1,000 object classes and over 1 million annotated images, and Common objects in context (COCO) with 80 object classes and about 220,000 annotated images (T.-Y. Lin et al., 2014). The pre-trained networks are generally used as network backbones in many CNN models such as YOLO with Darknet (Redmon et al., 2016) and Mask R-CNN with ResNet (K. He et al., 2017). Since the pre-trained network has been trained on a large amount of data, the use of a transfer learning approach to modify the deep learning model is not as long and does not require as much manual intervention as building a deep learning model from scratch (O'Mahony et al., 2019). On the other hand, the feature learning and extraction ability of the pre-trained model can be used in the network backbone by adding modifications through fine-tuning (Gulli & Pal, 2017). Therefore, the use of transfer learning and fine-tuning is an efficient and accurate means of building deep learning models.

2.5.2.2 Training deep learning models

For supervised learning based CNN model training, it is essential to create an extensive annotated training images dataset. This requires high computational resources including graphic processing units (GPUs), high performing memory, a processor and storage (O'Mahony et al., 2019; Z. Wang et al., 2021). In addition to the hardware requirements, computing platforms such as compute unified device architecture (CUDA), and libraries such as CUDA based deep neural networks (cuDNN) should be installed for GPU enabled deep learning model execution (Jian et al., 2013; Jorda et al., 2019). Therefore, for the successful implementation of deep learning models, selecting a proper training platform is crucial. With the advances in modern computing systems, GPU-enabled gaming computers such as Alienware and MSI and embedded edge computing devices of single board and miniature size such as NVIDIA Jetson TX2, NVIDIA Jetson Nano, and Raspberry Pi 3B+ have been used to implement deep learning models (Pal & Hsieh, 2021). Despite these advances, the hardware requirements are still expensive, and the

configurations needed for training deep learning models are complicated and time consuming. For deep learning model training, a functional computer with the hardware requirements specified above currently costs approximately US\$2,000 (approximately AU\$3,000).

With the proliferation of cloud computing and virtualisation, leading technology companies have provided dedicated development environments to overcome these hardware and configuration issues that have been impeding deep learning model deployment (Ekanayake, 2022). Examples at the forefront of this development include Colaboratory (Colab) by Google, Azure Machine Learning by Microsoft, Watson Studio by IBM, and SageMaker by Amazon (Pal & Hsieh, 2021). Virtualisation using cloud computing is the process of creating a virtual version of a physical computer with a dedicated amount of processor, memory and GPU borrowed from a cloud provider's server. As a result of this, virtual machines remain independent of the local physical host computer (Rahman et al., 2022). Another major advantage of using virtual machines to develop deep learning models for construction applications is the efficient deep learning model sharing among project stakeholders in the cloud without configuration modifications (Pal & Hsieh, 2021). Successful use of the virtual machine platform, Google Colab is evident in the studies of Canesche et al. (2021), Carneiro et al. (2018) and Ohkawara et al. (2021).

The onerous process of building and training deep learning models using extensive computational resources outweighs the anticipated benefits of deep learning models. To be able to realise the benefits of deep learning models for indoor construction progress monitoring, transfer learning and fine-tuning based deep learning model building and training using virtual machine platforms can be adopted.

2.5.3 Calculating work-in-progress using the deep learning model

The deep learning model should be capable of image segmentation, in addition to detection and classification for indoor construction work-in-progress calculations. Therefore, it has been identified as a research gap. Image segmentation using deep learning can determine the boundary of an object using a pixel mask and detect and classify the object with a bounding box simultaneously (W. Wei et al., 2022; Ying & Lee, 2019). As per semantic segmentation, multiple objects of a single object class are treated as one entity. In instance segmentation, multiple objects of the same class are considered

as distinct individual instances. Due to the generation of pixel masks for individual instances, instance segmentation is effective in revealing the presence, shape, size, count and location of the objects (Casado-García et al., 2019). However, under supervised learning, labelling the images with pixel level annotation is a time consuming and tedious task (Bharati & Pramanik, 2020). Deep learning models have been developed for two types of image segmentation methods: semantic segmentation and instance segmentation (Khoreva et al., 2017).

The common deep learning based segmentation algorithms include DeepMask (Pinheiro et al., 2015) and Fully Convolutional Instance-aware Semantic Segmentation (Li et al., 2017). However, these two algorithms are not suitable for segmenting overlapping objects. Moreover, for segmentation tasks, U-Net (Ronneberger et al., 2015), LinkNet (Chaurasia & Culurciello, 2017), Pyramid Scene Parsing Network (PSPNet) (Zhao et al., 2017) and Mask R-CNN (K. He et al., 2017) are the other prominent algorithms that are currently used. You Only Look at Coefficient++ (YOLOCT++) (Bolya et al., 2019) and YOLOMask (Hua et al., 2021) are the latest segmentation algorithms. With its high detection accuracy and instance segmentation performance, Mask R-CNN is considered a powerful segmentation algorithm and widely adopted in recent years (D. He et al., 2022; Xi et al., 2020). Compared to other segmentation algorithms, the instance segmentation ability is further used to solve computer vision issues such as pose estimation and key point detection (K. He et al., 2017). Mask R-CNN can accurately segment overlapping objects (Zhang et al., 2020) and is suitable for detecting objects with irregular shapes (W. Wei et al., 2022; Ying & Lee, 2019). An optimised Mask R-CNN model through transfer learning and fine-tuning can be post-processed for automatic calculation of work-in-progress ratios of indoor construction works.

Based on the research gaps, the three research questions are formulated as follows.

1. What are the technical challenges in indoor construction environments and how do they affect the automated visual recognition of as-built elements?
2. How can deep learning be employed for automated visual recognition of indoor as-built elements by reducing the computational complexities?
3. How can deep learning be employed to facilitate the automated calculation of work-in-progress of indoor elements?

A methodological framework has been devised in Chapter 3 to answer the research questions and they lead to answering the central research problem of how to improve automation in computer vision based indoor construction progress monitoring. Thereby to achieve the aim of developing a deep learning based approach for improving automation in computer vision based indoor construction progress monitoring.

2.6 Chapter summary

This chapter reviewed the progress monitoring methods in construction with an emphasis on the challenges of traditional manual and laborious methods. Through this chapter, the state-of-the-art technologies for automated construction progress monitoring were presented. It was emphasised that computer vision technology has competitive advantages. The algorithmic approaches to analyse the as-built status were reviewed with an emphasis on conventional machine learning algorithms, which rely on manual feature extraction. Performing robust object detection, classification and segmentation by deep learning algorithms has been crucial to document and analyse the as-built state of recent computer vision based construction progress monitoring studies. This chapter also reviewed the applications of computer vision based technologies for indoor construction progress monitoring and their advances over the past decade. These studies have predominantly used manual feature extraction using machine learning algorithms. The recent developments in computer vision based indoor construction progress monitoring have harnessed the detection, classification, and segmentation ability of Mask R-CNN for recognising indoor elements. Through the extensive literature review conducted in this chapter, three research gaps were highlighted.

Considering the limitations of computer vision based indoor construction progress monitoring studies, it was discerned that it is essential to determine the impacts of technical challenges for as-built indoor elements visual recognition. A review of these technical challenges was presented in section 2.5.1 as a research gap. Chapter 4 explains how the empirical evidence obtained in the form of indoor site images from the case projects is used to further demonstrate the challenges and their impacts for indoor as-built visual recognition. With the recent applications of deep learning models to minimise the impacts of technical challenges, the need to develop a simplified, efficient, and shareable deep learning approach to improve as-built indoor elements visual recognition was highlighted as a research gap. Therefore, the challenges related to deep learning models

and the means to overcome them were reviewed in section 2.5.2. The research gap on employing a segmentation deep learning model with the detection and classification abilities for automated and accurate calculation of indoor work-in-progress was reviewed in section 2.5.3. A methodological framework is presented in Chapter 3 to answer the research questions, which leads to answering the central research problem and achieving the aim of this study.

3 Research design and methods

To present the research design and methods, this chapter has three main parts. In the first part, the chapter focuses on determining the methodological framework. Section 3.1 presents the design science research framework that has been selected as the methodological framework. The process of answering the research questions has been mapped with the design science research framework. Section 3.2 presents the process of data collection from the case projects and explores the data collection process with reference to the case projects and the creation of the training images dataset. Section 3.3 investigates the research design for developing the deep learning based approach. This section provides a detailed discussion on Mask R-CNN model's neural network architecture, the methods to build and evaluate deep learning models and the use of Colab virtual machine as the deep learning model training platform. Section 3.4 summarises the research process of this study.

3.1 Determining the methodological framework

Choosing the right methodological framework is guided by the nature of the research problem (Saunders et al., 2009, p.173) and the aim of the study (Yin, 2009, p.40). The nature of the research problem and the aim indicates that this study focuses on addressing a practical problem by developing an artefact. The design-oriented framework embodies the process of understanding a practical problem, designing, and developing the artefact, evaluating the performance of the artefact, and communicating the results (Hevner et al., 2004; Iivari & Venable, 2009). The design-oriented framework is incorporated in constructive research approach, action research and design science research.

The constructive research approach is widely known for providing a design-oriented framework for management research (Jönsson & Lukka, 2006; Lukka, 2003). Although it follows the design-oriented framework, the constructive research approach is mostly dedicated to expanding the relevance of management science research by using theory to design constructions (Lukka, 2003). When developing an artefact, the constructive research approach prioritises the collaboration with practitioners from the target organisation by setting up a joint project team (Jönsson & Lukka, 2006; Piirainen & Gonzalez, 2013). Market research is the most common testing and evaluation method of the artefact designed (Piirainen & Gonzalez, 2013). There is ample evidence of employing the constructive research approach adhering to the design-oriented framework

in management applications. Among such examples, Kasanen et al. (1993) created a model for capital budgeting in an organisation, while Lindholm (2008) devised a corporate real estate management framework using the constructive research approach.

Although it is often treated as a research strategy (Iivari & Venable, 2009), action research embodies the design-oriented research framework for developing an artefact to solve a practically relevant problem (Järvinen, 2007). Action research is characterised by the researcher engaging in the social system, which is the problem context (Baskerville & Myers, 2004). Action research incorporates a design-oriented framework to address problems and devise solutions related to complex workings of organisational situations and human activities (Avison et al., 2001). In such situations, action researchers work with practitioners to solve the important practical problem by immersing themselves in the organisational settings (Iivari & Venable, 2009; Järvinen, 2007). Avison et al. (2001) and Baskerville (1999) investigated how action research can be employed using the design-oriented framework for database prototyping in organisations by collaborating with the organisational staff.

3.1.1 Design science research

As the widely used design-oriented research framework, **design science research** prioritises the understanding of a practical problem and devises its solution by developing an artefact (Hevner et al., 2004; Vaishnavi, 2007). An essential difference between design science research, constructive research approach and action research is that design science research does not presume a specific client or a collaboration between the researchers and the practitioners (Iivari & Venable, 2009). This is mainly because design science research intends to offer an open framework for design-oriented research (Baskerville, 2008) without being only limited to a single organisation. Compared to the constructive research approach, design science research recognises other forms of validation besides a market test (Piirainen & Gonzalez, 2013). Although action researchers are required to immerse themselves in the problem context, design science research does not necessitate such engagements (Baskerville & Myers, 2004; Iivari & Venable, 2009). Design science research contributes to producing technology based artefacts to resolve a practical problem, compared to the constructive research approach and action research (Piirainen & Gonzalez, 2013). Considering the flexibility and adaptability of design science research to develop a technical artefact for a practically relevant problem, design science research

has been selected as the appropriate design-oriented research framework to address the research problem and to attain the aim of this research.

As design science research has its roots in engineering and information technology (Hevner et al., 2004), the design science research framework was originally used to devise artefacts in information systems related research (Van Aken et al., 2016). In design science research, the artefacts can be classified into constructs, models, methods, and instantiations (Hevner et al., 2004). These artefacts encompass creations such as software solutions, algorithms, and mathematical models (Alter, 2008; Benbasat & Zmud, 2003). Among the previous research on employing the design science research process in practice, Krawatzeck et al. (2013) introduced a software intensive method for computer-aided data warehouse engineering in an organisation and Bodenbenner et al. (2013) designed an electricity demand response system. When developing artefacts for construction progress monitoring related problems, Martinez et al. (2021) employed the design science research framework to devise a solution for monitoring floor panel installations using a R-CNN based model.

3.1.2 Research process mapped with the design science research framework

With the intention of operationalising the design science research framework, Vaishnavi and Kuechler (2004) introduced a design science research process. Peffers et al. (2007) standardised this design science research process by adapting it as a methodological framework for design science research. This widely used design science research framework (Johannesson & Perjons, 2014; Peffers et al., 2007; Vaishnavi & Kuechler, 2004) is illustrated in Figure 3.1.

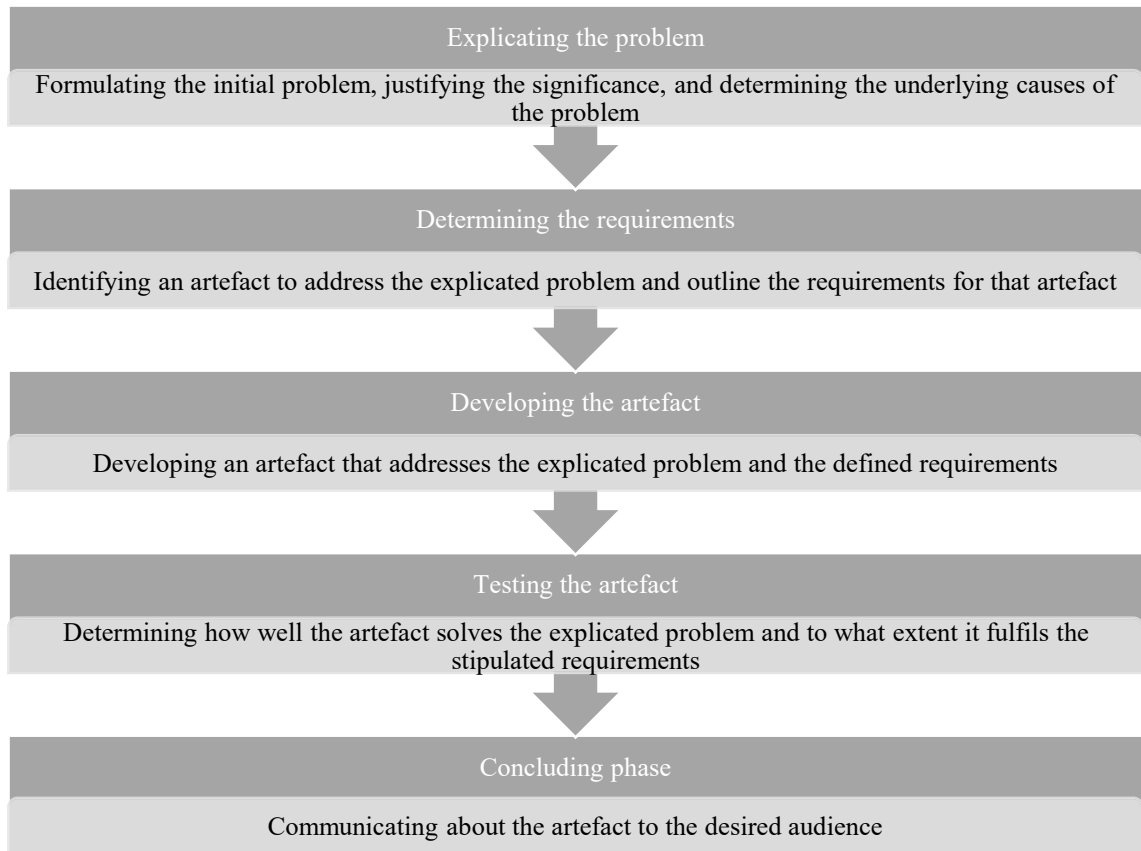


Figure 3.1: Process of design science research

Source: Johannesson & Perjons (2014); Peffers et al. (2007); Vaishnavi & Kuechler (2004)

As depicted in Figure 3.1, the design science research framework adapted for this study is inspired by the information systems related design science research studies of Johannesson and Perjons (2014), Peffers et al. (2007) and Vaishnavi and Kuechler (2004). In the subsequent subsections, the design science research framework is mapped with the research process of developing a deep learning based approach for improving automation in computer vision based indoor construction progress monitoring. The research methods to fulfil the objectives of this study are investigated in the research process.

The first phase of the design science research framework is explicating the problem. This phase aims to formulate the initial problem, justify the significance, and determine the underlying causes of the problem. A problem is initially formulated by a systematic review of past studies. In order to justify the significance of the formulated problem, it has to be positioned in the practical context. A more detailed understanding of the problem is assimilated by determining the underlying causes of the problem in the context

(Johannesson & Perjons, 2014). The research gaps of this research were determined from the systematic literature review and these gaps are discussed in section 2.5.1, section 2.5.2, and section 2.5.3. To understand the practical problem in the real-life context, three indoor construction sites are used as the case projects.

The second phase of the design science research framework intends to identify an artefact to address the explicated problem and outline the requirements for that artefact. In design science research, when defining requirements, researchers usually have an artefact in mind for which they can stipulate the requirements (Johannesson & Perjons, 2014; Peffers et al., 2007; Vaishnavi & Kuechler, 2004). As discerned from the research gaps, the outlined solution for the explicated problem of improving automation in computer vision based indoor construction progress monitoring is a deep learning based approach built on indoor construction progress images and a Mask R-CNN model. The deep learning model is trained and shared on a Google Colab virtual machine.

The third phase of the design science research framework focuses on developing an artefact that addresses the explicated problem (Johannesson & Perjons, 2014; Peffers et al., 2007; Vaishnavi & Kuechler, 2004). The specific performance capabilities of the artefact are developed during this phase by fulfilling the defined requirements (Gregor & Hevner, 2013; Hevner et al., 2004). When employing a deep learning model in the indoor construction progress monitoring context, enhancing the performance of the deep learning model is imperative to minimise the impacts of technical challenges in automated visual recognition. The Mask R-CNN model is optimised to achieve high mAP and low training and validation loss, such that the deep learning model's accuracy is high when recognising indoor elements in the images affected by background noise and lighting conditions. The segmentation ability of the optimised Mask R-CNN model is used to calculate the pixel area of the segmented masks in the site images. The pixel area is used to calculate the progress completion percentage of the indoor works.

In the phase of testing the artefact, researchers ascertain how well the artefact solves the explicated problem and to what extent it satisfies the stipulated requirements. Depending on the performance requirement, testing the artefact could be a demonstration or an implementation of the artefact. The bare minimum requirement is to demonstrate that the artefact works in one illustrative or real-life case (Piirainen & Gonzalez, 2013). The design science research process moves to the final phase only after confirming that the

artefact is stable and satisfies the stipulated requirements (Hevner et al., 2004; Piirainen & Gonzalez, 2013). The deep learning model is trained on the images collected from two case projects and is tested on the site images collected from a third indoor site.

After confirming that the artefact meets the stipulated requirements, the fifth stage of the design science research framework is the concluding phase, where the results are communicated to the desired audience. Since the present study does not intend to deliver a commercial product, this phase of the design science research framework is not applicable for the research process. For research and an academic audience, the results of this study are communicated mainly through academic journals and conferences.

The design science research framework which embodies the process of explicating the problem, determining the requirements for the artefact, developing the artefact, testing the artefact, and communicating the results was mapped with the research process of developing a deep learning based approach to improve automation in computer vision based indoor construction progress monitoring. The research process is initiated with data collection from the case projects.

3.2 Data collection from the case projects

Researchers have particularly used case projects to demonstrate, develop, and test prototypes in real-life situations (Fellows & Liu, 2015, p.161; Stake, 2013, p.23). In automated construction progress monitoring studies, the use of case projects for site images collection is the standard practice for algorithmic models training and demonstration (Deng et al., 2020; Hamledari, McCabe, & Davari, 2017; Kropp et al., 2014; Kropp et al., 2018; W. Wei et al., 2022; Wong et al., 2022).

3.2.1 Case projects

Three indoor construction sites were used as case projects to collect progress images to train and demonstrate the performance of the deep learning model. All the sites were located in the state of New South Wales (NSW), Australia. The Human Ethics Research Committee of University of Technology Sydney approved the ethics application to conduct the data collection. The approval (ETH20-5459) is presented in Appendix 1. These selected case projects, which were of small and medium scale belonged to a single building contractor organisation. The written approval of the Project Manager of these projects were obtained for collecting the images of the selected progress monitoring

scenario using fixed cameras. It was ensured that the identity of the construction workers captured during the progress images capturing process could not be recognised in the images. The progress images, 3D construction drawings and models were obtained ensuring that there was no potentially identifiable information about the project and the building contractor.

The systematic literature review uncovered the technical challenges that impede automation for computer vision based as-built indoor elements visual recognition. To understand the practical problem in the real-life context, where the technical challenges practically occur, indoor construction sites were used as the case projects capturing the process of framing, insulation, and drywall installation of indoor partitions in time-lapse images. Capturing an in-depth understanding of the technical challenges and their impacts for computer vision based as-built indoor elements visual recognition with reference to the case projects contributed to corroborate and modify the literature findings. While understanding the challenges and their impacts, two indoor construction sites were also used to capture the training images for the optimised Mask R-CNN model and the third site was used to capture the images to test this deep learning model. In addition to as-built site images, 3D construction drawings and models were collected from the selected sites. Some images captured from the indoor sites in the three Sydney suburbs of Alexandria (Project 1), Mosman (Project 2) and Bronte (Project 3) are presented in Figure 3.2a, 3.2b and 3.2c respectively. The details about these case projects are summarised in Table 3.1.

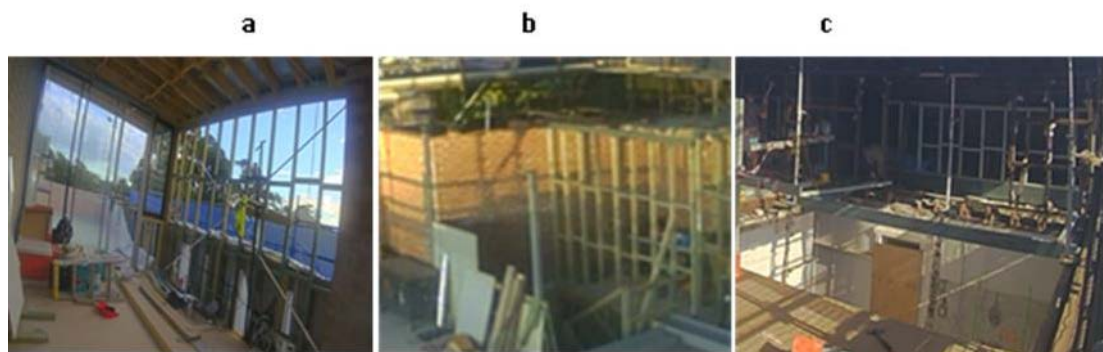


Figure 3.2: Case projects: a) Project 1; b) Project 2; c) Project 3

Table 3.1: Information about the case projects

	Project 1	Project 2	Project 3
Nature of the project	Residential building project of a townhouse	Commercial building project with fit-out work for office workspace	Alterations and additions to an existing dwelling
Location of the site (City/Suburb)	Alexandria, NSW	Mosman, NSW	Bronte, NSW
Number of floor levels	2	3	2
Duration of data collection	20 April 2021 – 9 July 2021	14 October 2021 – 26 January 2022	30 March 2022 – 20 May 2022
Number of images from the progress monitoring scenario that were used in this study	703 images for the training dataset	675 images for the training dataset	100 images for testing

Based on previous studies such as Deng et al. (2020); Hamledari, McCabe and Davari (2017); Kropp et al. (2014) Kropp et al. (2018); and W. Wei et al. (2022), it is evident that the number of case projects and the nature of the indoor construction has no impact on training images if a sufficient amount of images has been captured to understand the features of the objects. In previous studies, which employed deep learning for recognition of indoor objects, Ying and Lee (2019) used four buildings of a university collecting 1,076 training images, and W. Wei et al. (2022) used a building in a community project collecting 500 images for training. In the current study, 1,378 training images were collected from the two sites. As noted in section 1.5, during the data collection period of April 2021 to May 2022, the case projects were heavily affected by COVID lockdowns and restrictions imposed in NSW. As a result of the COVID outbreaks in NSW, construction activities across NSW were temporarily shut down on 19 July 2021 and work resumed for vaccinated workers from 11 August 2021 (Master Builders Association New South Wales, 2021). With the COVID outbreaks, construction personnel being infected and material supply issues, the construction works in Project 2 and Project 3 were delayed.

3.2.2 Time-lapse cameras, related accessories, and camera settings

To capture image data in the case projects, two time-lapse cameras (Brinno TLC200 Pro) were used as fixed cameras at each site. Figure 3.3 shows the front (camera lens) and the back (settings) of the Brinno TLC200 Pro.



Figure 3.3: Brinno TLC200 Pro time-lapse camera

The reason for choosing time-lapse cameras is their ubiquitous use in construction sites. They are commonly used to record progress for monitoring, marketing, and security purposes, making the computer vision based approaches more practical and feasible (J. Kim et al., 2018). The same popular and inexpensive technology can be employed for monitoring a chosen as-built scenario, while fixed at a vantage point with the least manual intervention (Ahmadian Fard Fini et al., 2022). The dimensions of the Brinno TLC200 Pro are 6.4 x 5.2 x 10.7 centimetres, and the weight is 140 grams without the batteries. The camera is equipped with a waterproof and weather resistant housing, which can be used to cover the camera and install it with ease. These specifications prove that the camera is lightweight and easy to handle in congested indoor sites. As of 2022, the camera was priced at approximately US\$220 (approximately AU\$ 330).

Although the literature findings revealed that the use of a fixed camera leads to lack of field of view and relocation, the use of UAVs or UGVs could pose more challenges as indoor sites are congested. The safety of the workers, and potential disturbance to workers with the operation of UAVs and UGVs, are reasons not to use them for data collection. In addition, as per the legal regulations for flying drones in Australia, the Civil Aviation Safety Authority states that the drone must be kept at least 30 metres away from other people and the drone must not fly over or above people in occupied areas (Civil Aviation Safety Authority, 2022). These conditions are impractical to meet in a cluttered and enclosed indoor site, while construction workers are performing construction operations. Use of a web camera (webcam) could facilitate real-time data capturing when the researcher is offsite. A time-lapse camera could be connected through a secured internet connection to function as a webcam. However, there was no stable internet connection at

the selected indoor sites and setting up the webcam connection would have incurred an additional cost. Therefore, the Brinno TLC200 Pro time-lapse camera was used as a fixed camera at the best vantage point for the duration of recording the selected activities. The camera was moved in the site when the construction activities of the same progress monitoring proceeded in different locations of the site. Using fixed time-lapse cameras proved to be more practical during the global COVID-19 pandemic due to the rules of social distancing and minimal contact. Figure 3.4 illustrates how cameras were installed at the best vantage points considering the availability of the contractor's resources, site facilities and surrounding structures.

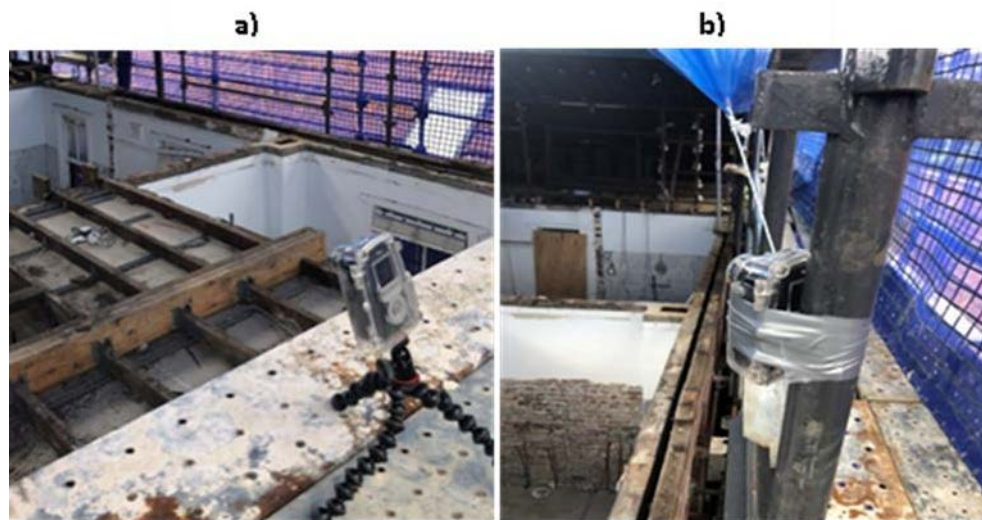


Figure 3.4: Installing cameras: a) Using a tripod; b) Taping to scaffolding

In some vantage points, a tripod was used to support the camera view and the position. This specialised tripod, called the “Gorillapod”, has a competitive advantage over a traditional tripod in dynamic indoor environments with its flexibility, portability, and ability to be attached to a non-horizontal surface. Figure 3.4a illustrates how the camera was supported with a Gorillapod set up on scaffolding. In Figure 3.4b, the camera was taped to scaffolding to gain the necessary height to capture the best viewpoint. To protect the camera in the cluttered indoor sites, a weather resistant housing was used. Figure 3.5a and Figure 3.5b show the Gorillapod tripod and the weather resistant housing respectively.

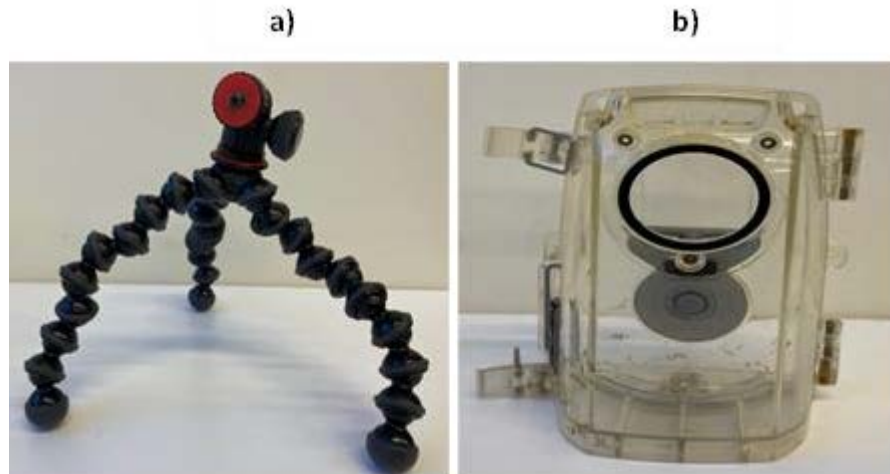


Figure 3.5: Camera accessories: a) Gorillapod; b) Weather resistant housing

The reason for using two cameras is to capture images from two different lighting perspectives. Indoor construction sites use both artificial lights inside and natural sunlight seeping through openings. Therefore, one camera was located to monitor the indoor wall in an area with artificial lighting, while the other camera was set up facing the indoor wall at the site, with sunlight seeping into the indoor location. Figure 3.6 depicts how cameras were set up at two different locations in Project 1 with different lighting conditions.



Figure 3.6: Indoor location with a) Natural lighting; b) Artificial lighting

Camera settings were adjusted to accommodate the indoor conditions. The sensor of the camera is High Dynamic Range (HDR) with 115dB and 1/3 inches. This lens has a resolution of 1.3 Mega pixel and a pixel size of 4.2 μm enabling a detailed view in low light situations. It has a field of view of 112°. The resolution of the images captured is 1280×720 pixels. The time-lapsing video recording format is Audio Video Interleave (AVI) (Brinno, 2022). In the menu of the camera, the settings can be adjusted according to the scene the camera is capturing. A summary of these settings is tabulated in Table 3.2. Time lapsing of the frames was set to 1 minute with 30 frames per second (FPS) to

facilitate a wide volume of overlapping images and to avoid missing sensitive details of the as-built progress. The daily timer was set from 7 am to 4 pm to coincide with the construction routine operation of the selected sites from 7 am to 3 pm in Project 1 and 7.30 am to 3.30 pm in Project 2. Project 3 was operating from 7.30 am to 3.30 pm. Although the sites were operating only on weekdays from Monday to Friday, cameras were still in operation on Saturdays and Sundays. Since the images were captured during the daytime, scene was set to daylight mode. The mode of capturing was set to low light. In the daylight scenery and low light capture modes, the white balance was set to auto mode to adjust the colour balance of lighting. This ensures that the lighting appears neutral white, counteracting the effects of artificial lighting sources. The timestamp was switched on to collect the meta data on the dates and times for as-built work-in-progress calculations.

Table 3.2: Time-lapse camera settings

Parameter	Camera settings
Time-lapsing interval	1 minute apart
Capturing speed	30 FPS
Low light mode	on
HDR range	high
Scene	daylight
Timer	7 am to 4 pm
Image quality	best
Timestamp	on
White balance mode	auto

The camera is supported by four batteries which have a battery life of approximately 6 to 8 weeks. A Secure Digital (SD) card of capacity 32GB was used as the memory storage. As the construction progressed, when the cameras needed to be moved to a different location in the site to capture the progress monitoring scenario, it was coordinated with the project manager of each site. The cameras were checked, and videos were collected on a fortnightly basis. Martinez et al. (2019) note the importance of these regular check-ups to avoid memory and battery outage and damage to the cameras due to the heavily cluttered and occluded indoor sites.

3.2.3 Developing the training images dataset

To collect the training images from Project 1 and Project 2, the time-lapse videos obtained in AVI were converted to images in Joint Photographic Experts Group (jpg) format using a video to image converter software (Video to jpg Converter). The use of jpg format is mandatory as it is the accepted image format for the Mask R-CNN training process (K. He et al., 2017). For supervised learning, it is essential to build an extensive training image dataset with annotation of target objects (Fang et al., 2018; L. Liu et al., 2020; W. Liu et al., 2019). To create a rich dataset of diverse images from each of the activities in the framing, insulation and drywall installation scenario, publicly available online images were used in the training process in addition to the images captured from Project 1 and Project 2. Sourcing images from the internet to build training image datasets has been carried out in previous studies, which used deep learning models (Slaton et al., 2020; Ying & Lee, 2019; W. Wei et al., 2022; Z. Wang et al., 2021) for object recognition in construction site images. The images sourced from the internet which had distinctive colours, texture, material of the indoor framing, insulation, and drywall elements and different levels of illumination, occlusions, and clutter captured with different angles added diversity to the current image dataset collected from the case projects. Having a diverse image dataset ensures that the deep learning model is able to understand the patterns for feature extraction by overcoming the underfitting problem (Jabbar & Khan, 2015). The problem of underfitting of deep learning models relates to the failure to capture relevant patterns, which leads to less accurate predictions (Jabbar & Khan, 2015). As shown in Figure 3.7(a), the image was captured from insulation in Project 1 and the image in Figure 3.7(b) was sourced from insulation images available online. Sourcing images from the internet further ensured that the deep learning model is not overfitted to the images captured from Project 1 and Project 2.

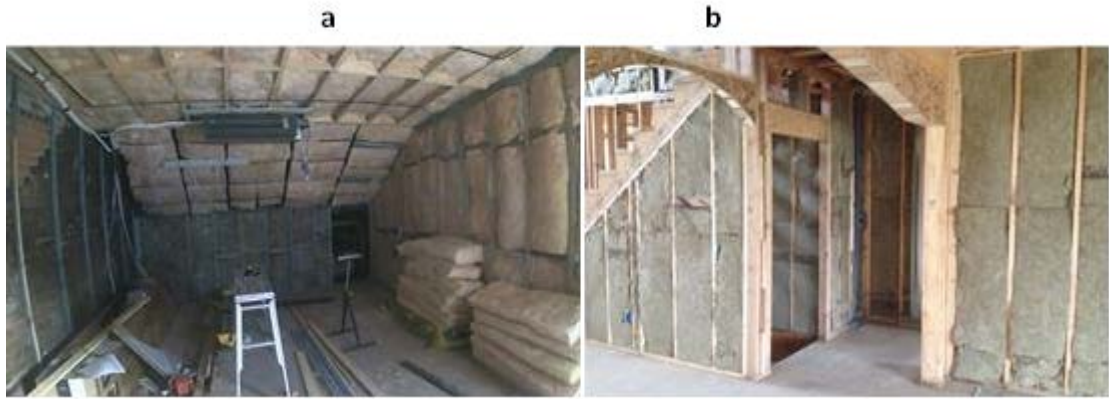


Figure 3.7: a) Image captured from Project 1; b) Image sourced from the internet

A total of 1,378 (703+675) images were collected from the two case projects. Representing the same progress monitoring scenario, 422 images were sourced from the internet. The images from the sites and the images sourced from the internet dataset comprised 1,800 images. Past studies, such as Slaton et al. (2020) and Z. Wang et al. (2021) which used a similar approach, emphasise that there is no absolute ratio for the images sourced from the internet and the construction sites. Sun et al. (2017) and Soekhoe et al. (2016) point out that the number of images in the training dataset has no fixed value, but highly recommend having a few thousand images depending on the complexity of the targeted scenario to ensure the deep learning model can learn the features. The accuracy of the deep learning model increases, when the number of high quality images in the dataset increase (J. Wang et al., 2018).

To generate more diversified training images and to further minimise the overfitting problem, data augmentation was applied. Overfitting in deep learning models occurs when the model accurately recognises objects in the images that were included in the training set but is not as accurate at recognising objects in the images that it was not trained on (Rice et al., 2020). Data augmentation is a technique to transform the existing images to create new versions of the original images (Shorten & Khoshgoftaar, 2019). Data augmentation can be performed through photometric distortions and geometric distortions. Adjusting the hue, saturation, brightness, contrast, and noise of an image are examples of photometric distortion. Strategies for geometric distortion are random cropping, flipping, scaling, and rotating (Bochkovskiy et al., 2020). Using the machine learning library “imgaug” (imgaug, 2022) which is commonly used for image augmentation, a code was developed to enable image augmentation. Open source computer vision (OpenCV) is a prominent library of programming functions for such

tasks (imgaug, 2022). OpenCV version 4.6 was used to develop this code in the Jupyter Notebook. The code used for image augmentation is shown in Figure 3.8.

```
# Importing the libraries
import imgaug.augmenters as iaa
import cv2
import glob

# Extracting the images that are subjected to augmentation
images = []
images_path = glob.glob("images/*.jpg")
for img_path in images_path:
    img = cv2.imread(img_path)
    images.append(img)

# Implementing augmentation commands
augmentation = iaa.Sequential([
    #Rotate
    iaa.Rotate((0,20 )),
    #Flip the images
    iaa.Fliplr(0.5),
    iaa.Flipud(0.5),
    #Affine
    iaa.Affine(translate_percent={"x": (-0.6, 0.6), "y":(-0.6, 0.6)}),
    #Multiply
    iaa.Multiply((1.5, 1.5)),
    #Linear contrast
    iaa.LinearContrast((1.3, 1.3))
    #Sharpen images
    iaa.Sharpen(alpha=(0, 1), lightness=(1, 1)),
    #Emboss images
    iaa.Emboss(alpha=(0, 1.0), strength=(0, 2.0)),
    #Gaussian blur
    iaa.GaussianBlur((0.0, 4.0))
])

# Displaying the augmented images
while True:
    augmented_images = augmentation(images=images)
    for img in augmented_images:
        cv2.imshow("Image", img)
        cv2.waitKey(0)
```

Figure 3.8: Sample of Python code for image augmentation

According to this code, initially the programming libraries needed are imported. Then the images, which are required to be augmented are contained in a separate folder, such that they can be extracted to an array. In the next step, augmentation commands, “rotate”, “flip”, “affine”, “multiply”, “linear contrast”, “sharpen”, “emboss” and “gaussian blur”

are executed. Threshold values for each of the augmentation commands are assigned and optimised based on the requirements of each image. For the current research, all the commands were not executed at once. Individual commands were selected based on the requirements of each image. An example of the image transformation using augmentation commands is shown in Figure 3.9.

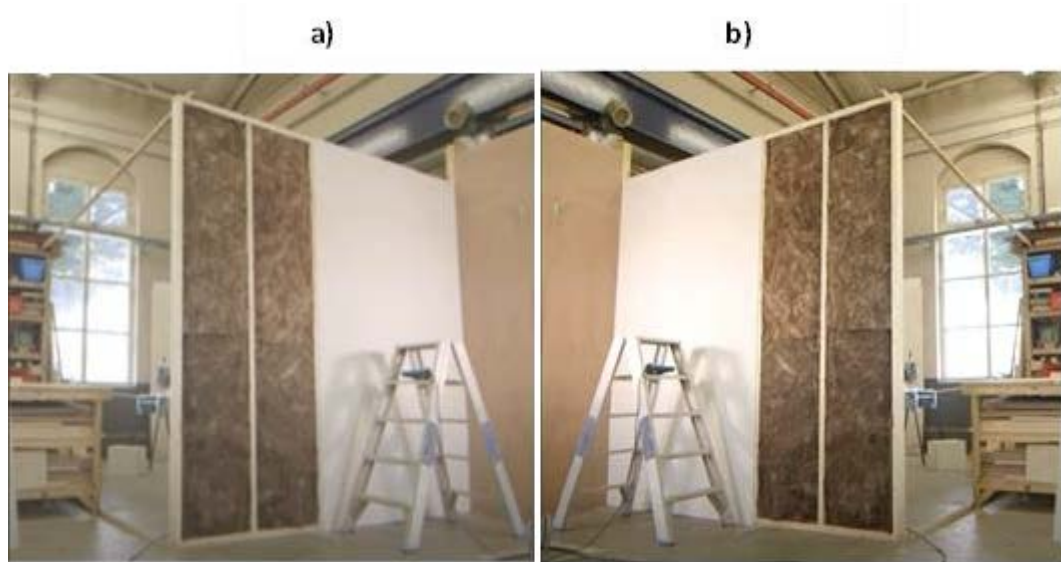


Figure 3.9: a) Original image; b) Image flipped vertically and contrast enhanced

As shown in Figure 3.9a, the original image, which was captured in a low lighting environment, was photometrically enhanced using contrasting and the image was geometrically diversified by vertical flipping. An additional 200 images from the image dataset were randomly augmented. In total, the training image dataset comprised 2,000 images. The images in each activity class of the dataset were labelled using the online annotation tool known as “Make Sense”. Figure 3.10 illustrates how the annotation was created for insulation with the label “drywall_installed”.

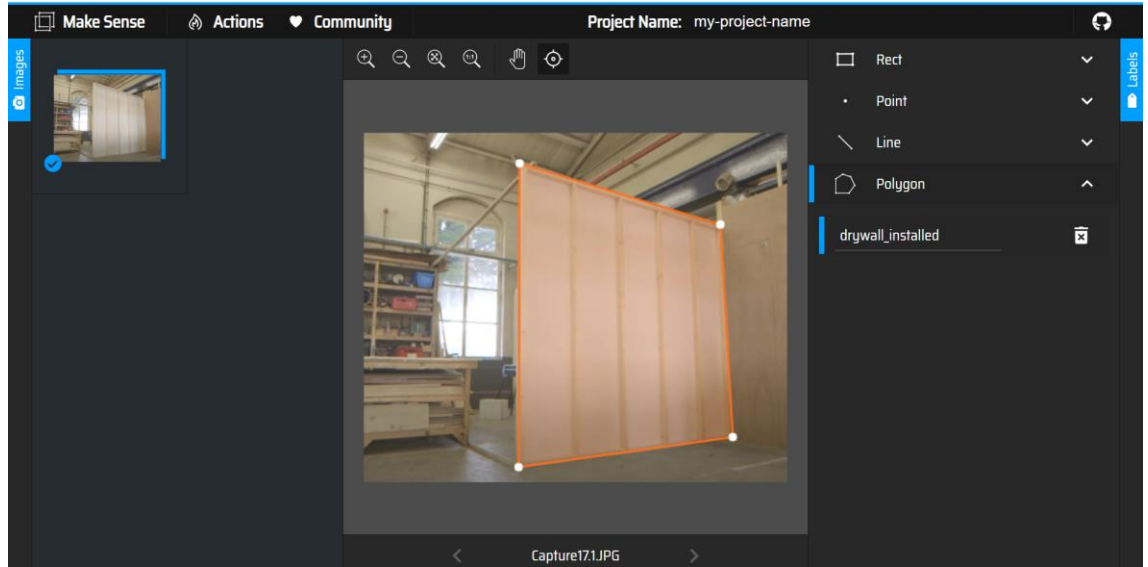


Figure 3.10: Annotated image

Each image had a polygon drawn over to segment the pixel area of the region of interest from the rest of the image. Thereby, the ground truth label was marked. The corresponding single file in Common Objects in Context JavaScript Object Notation (coco json) format was obtained to indicate the coordinates of the polygon. For each image, the information about the image path, width, height, and identification (ID) are included in the coco json format. The labels used in the dataset were “framed”, “insulated” and “drywall_installed”. The training image dataset was prepared with 2,000 annotated images.

3.3 Research design for developing the deep learning approach

The deep learning based approach presented in the current study is built on a Mask R-CNN model. As discussed in section 2.5.3, compared to other deep learning based segmentation algorithms, Mask R-CNN offers competitive advantages in terms of object detection, classification, and pixel level segmentation. The research design for building, training, and evaluating the Mask R-CNN model is navigated in the subsequent sections.

3.3.1 Mask R-CNN network architecture

Mask R-CNN was originally proposed by K. He et al. (2017), a group of “Facebook Artificial Intelligence Research” programmers. This model is an extension to the Faster R-CNN model (Ren et al., 2015) by inserting a branch to predict segmentation masks in addition to the existing branches to predict object classification and bounding box generation based detection. The mask branch is capable of creating a mask at pixel level

for each object detected (K. He et al., 2017). The network architecture of Mask R-CNN, with the key elements is illustrated in Figure 3.11.

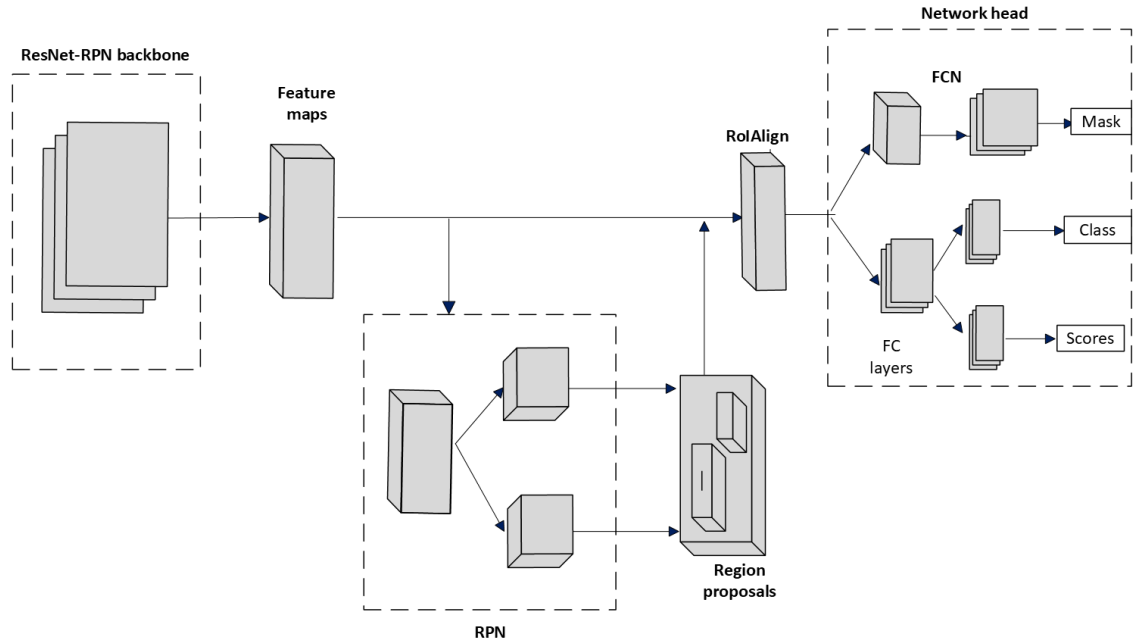


Figure 3.11: Mask R-CNN network architecture

Adapted from: K. He et al. (2017)

Each of the main components of the Mask R-CNN network architecture is discussed below.

- **The ResNet-FPN backbone** – This is the backbone architecture applied for feature extraction from the input image. The convolutional backbone for feature extraction in Mask R-CNN is based on Feature Pyramid Network (FPN) (Lin et al., 2017) and a ResNet architecture of networks of depth 50 or 101 layers (K. He et al., 2017; Kardovskyi & Moon, 2021). For smaller datasets, ResNet50 is recommended by K. He et al. (2017) because of the reduced time in training. However, ResNet101 is more accurate because of the high number of layers. The FPN utilises a top-down architecture with lateral connections to create a feature pyramid from a single-scale input (K. He et al., 2017).
- **The feature maps** – The ResNet-FPN backbone extracts the features from an input image and yields the feature map of the corresponding image (F. Wei et al., 2019).

- **Region Proposal Network (RPN)** – The RPN outputs the proposals for the regions by using the feature maps as the input (F. Wei et al., 2019).
- **Region of interest Align layer (RoIAlign)** – This layer ensures that the features extracted for each region of interest are well positioned with the regions of the original image (K. He et al., 2017; F. Wei et al., 2019).
- **The network head architecture** – The network head is responsible for bounding box recognition, class detection and mask prediction. A fully convolutional net (FCN) mask prediction branch is included in the head architecture, parallel to the object class and bounding box of detection branches of the fully connected (FC) layers (F. Wei et al., 2019).

The neural network of the Mask R-CNN model is built with different hyperparameters to fulfil the functions of each of the aforementioned crucial network elements. A hyperparameter is a parameter set by the programmer before the learning process commences. These hyperparameters can be fine-tuned depending on the requirement of training efficiency and model accuracy through optimisation (Goodfellow et al., 2016). Some of the crucial hyperparameters related to the training efficiency are batch size, number of iterations and network resolution. Some of the vital hyperparameters contributing to model accuracy through optimisation are learning rate, learning rate scheduler, optimiser, momentum, and decay (Chollet, 2017).

3.3.2 Building the deep learning model: Transfer learning and fine-tuning

In addition to FPN, Mask R-CNN uses ResNet as the backbone in the original model proposed by K. He et al. (2017). Since ResNet has been pre-trained on a COCO dataset of over 100,000 images (Howard et al., 2017), the use of transfer learning approach to modify the Mask R-CNN model is not as long and does not require as much manual intervention as building ResNet from scratch. On the other hand, the feature learning and extraction ability of the pre-trained ResNet can be used in the network backbone by adding customisations through fine-tuning (Gulli & Pal, 2017). Because of the learned parameters in the pre-trained model, the use of transfer learning also enhances accuracy (Nalini & Radhika, 2020). The transfer learning process is briefly illustrated in Figure 3.12.

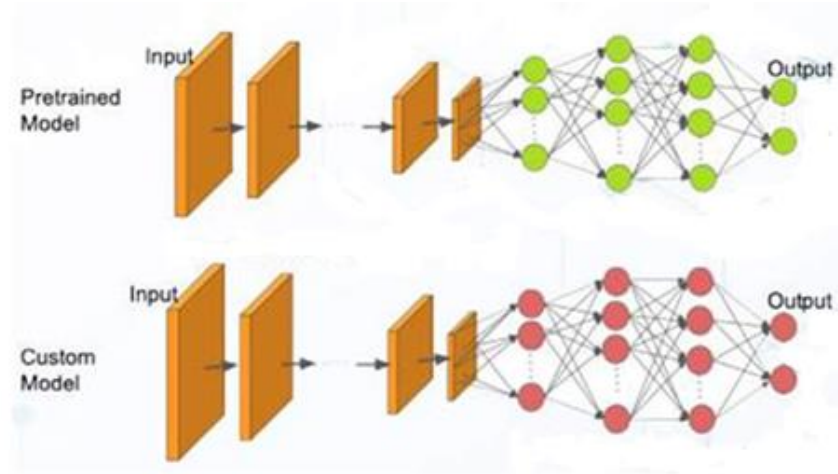


Figure 3.12: Illustrating transfer learning

Adapted from: Pan and Yang (2009)

As per Figure 3.12, a pre-trained deep learning model is developed with a certain set of hyperparameters in the neural network architecture (represented in green). Some of these original hyperparameters are optimised in the custom model (represented in red) through fine-tuning to improve accuracy. Previous examples of using transfer learning and fine-tuning in Mask R-CNN based construction object recognition and segmentation are evident in the studies of F. Wei et al. (2019); Guo et al. (2021); W. Wei et al. (2022) and Z. Wang et al. (2021). The original Mask R-CNN model by K. He et al. (2017) was trained on the COCO dataset of 80,000 images. When ResNet-50-FPN was adopted, the training process took about 32 hours, while the ResNet-100-FPN backbone resulted in a training process of 44 hours. A total of 8 GPUs was used for 160,000 iterations. The learning rate of the model, 0.02 was decreased by 10 at 120,000 iterations for optimum performance by a learning rate scheduler called “Multiple Steps learning rate (MultiStepLR)”. The default optimiser of Mask R-CNN is stochastic gradient descent (SGD). The weight decay is 0.0001 and the momentum is 0.9 (K. He et al., 2017). The building and training process of the optimised Mask R-CNN model using transfer learning and fine-tuning is elaborated in section 5.1 of Chapter 5.

To modify the original model, which is treated as the base model in transfer learning, certain hyperparameters need to be fine-tuned. Currently, there are no simple and easy ways to fine-tune the hyperparameters and select the exact set of hyperparameters (Smith, 2018). The programmers generally conduct many experiments to determine the optimum set of hyperparameters that improve the accuracy of the model. In certain scenarios, the

configuration files from freely available customisable deep learning models such as “model zoo” are employed (Feurer & Hutter, 2019). Examining the training and validation loss to overcome the underfitting and overfitting scenarios is another method to determine the optimal set of hyperparameters (Feurer & Hutter, 2019; Smith, 2018).

When using transfer learning and fine-tuning for building Mask R-CNN models, programmers use the Tensorflow or the PyTorch framework. The Matterport implementation of Mask R-CNN uses Tensorflow as the framework. One example of using Tensorflow as the framework is evident in the study of W. Wei et al. (2022) for indoor brick laying and plastering detection. The Facebook Artificial Intelligence Research team that is responsible for developing Mask R-CNN uses the Detectron2 GitHub repository with the PyTorch framework. One such example of PyTorch based Mask R-CNN training is evident in the study of F. Wei et al. (2019) for concrete crack detection. Choosing the deep learning framework to build the Mask R-CNN model depends on the programmer’s choice.

3.3.3 Training platform for the deep learning model: Colab virtual machine

After devising the method to build the deep learning model, the training of the deep learning model is a crucial step. As discussed in section 2.5.2, virtual machine technology has fostered efficient training platforms for deep learning models. Among the recent advances, Google Colab’s ability to run as a virtual machine with the runtime fully configured for deep learning model training and free-of-charge access to GPUs, memory and processors has gained widespread recognition (Canesche et al., 2021; Carneiro et al., 2018). In addition to the free version of Colab, Google offers Colab Pro, Colab Pro+ and Google cloud enabled platforms at a subscription (Google Research, 2022). Using Colab Pro for deep learning model training is by far the cheapest option as it saves money on special hardware requirements. Using Colab as a virtual machine requires only approximately US\$10 (AU\$15) per month and a Google account.

Colab is a web based “Jupyter” integrated development environment or a notebook enabled to execute Python codes. Colab notebooks are stored in Google Drive enabling Google Drive as the storage unit to be accessed from any web browser as opposed to using the hard drive in a local computer (Ekanayake, 2022; Ohkawara et al., 2021). Colab enables setting up a virtual machine as runtime by connecting to GPUs and tensor processing units (TPUs) hosted by Google or through Google cloud platform hosted

services. The Colab users can also opt to connect to a local runtime by executing the code in local computer's hardware (Google Research, 2022). Since zero configuration is required and most of the libraries are already set up, deep learning models can be trained on Colab with a few lines of code and can be shared, stored, and accessed using Google Drive (Ekanayake, 2022; Pal & Hsieh, 2021). A visual explanation of the working mechanism of Colab after Colab is connected to the virtual machine runtime is presented in Figure 3.13.

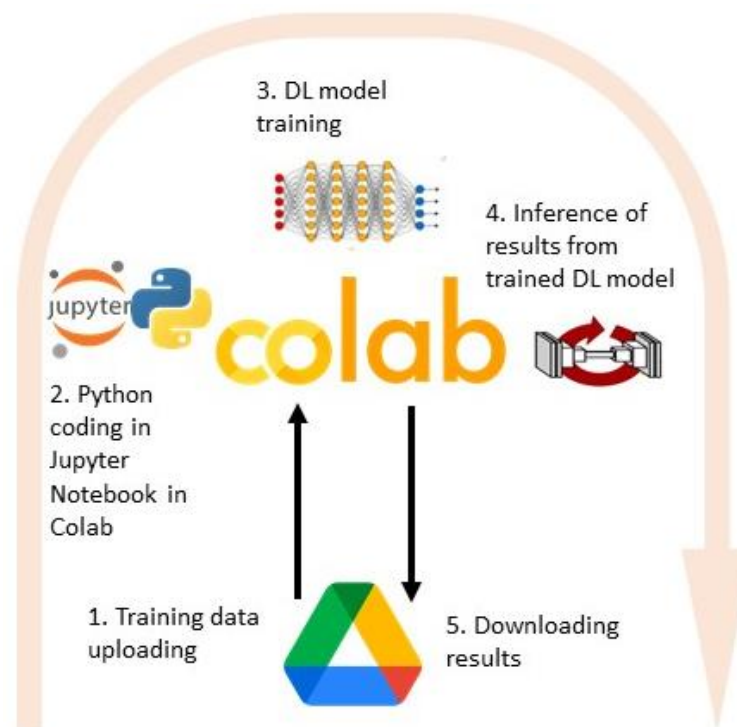


Figure 3.13: Working mechanism of Colab

Source: The Author

To train deep learning models on Colab, initially, the annotated training images with the annotation file are uploaded to Google Drive. Secondly, Google Drive is connected to the Colab notebook created from the same Google account. This is a Jupyter notebook enabled to execute the code for training the deep learning model, which is written in Python language. In the next step, the training command is executed. Because of the fast GPU and RAM performance in Colab, the training process is not as time consuming as a deep learning model training in a local computer. When the training is completed, the results are inferred using different metrics. Test images are uploaded to Google Drive for

inferring detection, classification, and segmentation results. Finally, the resulting images are downloaded from the Colab notebook.

3.3.4 *Metrics to evaluate the performance of deep learning models*

After the deep learning model is trained, certain metrics are used to evaluate the performance of the deep learning model. The current study focused on the **mAP** and the **loss** to capture the overall performance of the deep learning model at an intersection over union (IoU) of 0.5. The IoU measures how much the predicted boundary detected by the deep learning model overlaps with the real object boundary or the ground truth (Goodfellow et al., 2016). Accordingly, the deep learning model presented in this study does not detect objects, whose confidence of existence score is less than 50%. For a specific value of IoU, the average precision (AP) measures the precision/recall curve at recall values when the maximum precision value drops. The AP is calculated as the area under the curve by numerical integration. The mean average precision (mAP) is the average of AP over a range of IoU from 0.5 to 0.95 (O'Mahony et al., 2019). The metric mAP is widely used to evaluate the detection accuracy of deep learning models (Chollet, 2017; Goodfellow et al., 2016).

To comprehend how mAP is calculated, understanding the terminologies, precision and recall is essential (Chollet, 2017). The precision and the recall values are computed using true positives (TP), false positives (FP) and false negatives (FN) (O'Mahony et al., 2019). In this context, TP and FP are the number of objects correctly and incorrectly predicted, respectively, as the object of interest. True negatives (TN) and FN are respectively the number of objects correctly and incorrectly recognised as background (Hamledari, McCabe, & Davari, 2017). Precision indicates the proportion of positive identifications which are correct, while recall is a measure of the proportion of actual positives that are identified correctly. The mathematics behind Precision and Recall are presented in Equation 3.1 and Equation 3.2.

$$Precision = \frac{TP}{TP+FP} \quad \text{Equation 3.1}$$

$$Recall = \frac{TP}{TP+FN} \quad \text{Equation 3.2}$$

In addition to mAP, when training deep learning models, loss value indicates how well or poorly a certain model behaves after each iteration of optimisation. The reduction of loss after each or several iterations is an indication of the higher accuracy of the deep learning model (Akbari et al., 2021). When training a Mask R-CNN model, a multi-task loss on each sampled region of interest is defined as in Equation 3.3. The loss is defined as the sum of the classification loss (L_{cls}), the bounding-box loss (L_{box}), and the mask loss (L_{mask}) (K. He et al., 2017).

$$L = L_{cls} + L_{box} + L_{mask} \quad \text{Equation 3.3}$$

The classification loss corresponds to the loss assigned to improper classification of the object. The bounding box loss is the loss related to the localisation of the bounding box of the identified class. The mask loss is associated with the creation of masks on the identified objects (Guo et al., 2021). The mask loss is the most important loss since it is defined only on positive regions of interest. The mask target is the intersection between a region of interest, and it is associated with ground-truth mask. The loss derived from the classification loss, the bounding-box loss, and the mask loss is illustrated in terms of training loss and validation loss. Training loss is measured after each batch, while the validation loss is measured after each epoch. The impact of training loss and validation loss on the overfitting and underfitting scenarios of the deep learning model can be expressed as follows (Chollet, 2017; Goodfellow et al., 2016).

If validation loss $\gg$ training loss, the model is overfitting to a high extent.

If validation loss $>$ training loss, the model is overfitting to a certain extent.

If validation loss $\ll$ training loss, the model is underfitting to a high extent.

If validation loss $<$ training loss, the model is underfitting to a certain extent.

The model is considered as fitting if the training loss and validation loss are approximately the same value. The model is also considered to be perfectly accurate if a mAP of 100% is reported (Chollet, 2017; Goodfellow et al., 2016).

3.4 Chapter summary

This chapter discussed the underpinning methodological framework of this research, the process of data collection from the case projects and the research design for developing

the deep learning based approach. A summary of the research process adopted in this study is illustrated in Figure 3.14.

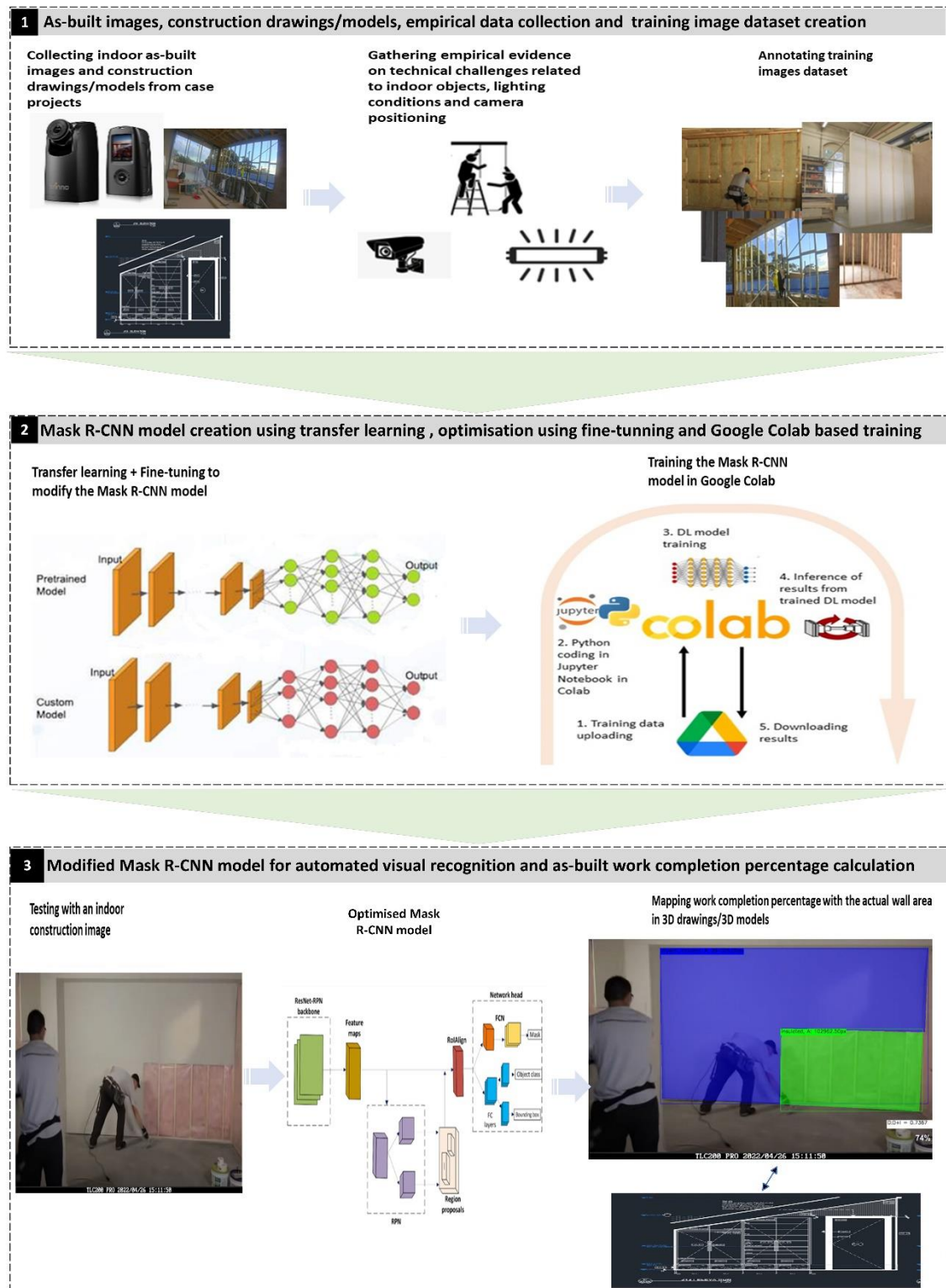


Figure 3.14: Summary of the research process

In step 1, data was collected in the form of site images of the progress of framing, insulation and drywall installation and construction 3D drawings and models. The case projects were used to corroborate and modify the literature findings on technical challenges and their impacts for indoor as-built visual recognition in Chapter 4. Two case projects were used to collect image data to train the deep learning model, which was built on an optimised Mask R-CNN model. As per step 2, the Mask R-CNN model was built using transfer learning and fine-tuning and trained on a Colab virtual machine platform. A third site was used to capture the site images to test the deep learning model in step 3.

This research process was mapped with the design science research framework. As per the design science research framework, achieving Objective 1, with reference to explicating the problem step, is presented in Chapter 4. Achieving Objective 2 and Objective 3 with reference to the subsequent steps in the design science research process, namely determining the requirements for the artefact, developing the artefact, and testing the artefact, is presented in Chapter 5.

4 Results and discussion: Part I

This chapter investigates how the technical challenges in indoor construction environments affect the visual recognition of as-built elements. Through the extensive review of studies on computer vision based indoor construction progress monitoring in section 2.5.1, the technical challenges related to computer vision based visual recognition of as-built indoor elements were classified into indoor objects, lighting conditions and camera positioning related challenges. To corroborate and modify the literature findings on the technical challenges and to further understand their impacts, the empirical evidence gathered during the process of image data collection from the three indoor case projects was used. The images captured from the case projects were used in section 4.1 to demonstrate the technical challenges and their impacts on feature extraction by using a handcrafted machine learning algorithm. Section 4.2 presents the pre-processing techniques to reduce the impacts of the technical challenges in the form of Python programming based coding extracts with explanations. Section 4.3 discusses the technical challenges and their impacts for visual recognition of as-built indoor elements. Section 4.4 summarises the chapter.

4.1 Empirical evidence from the case projects

The reason for gathering empirical evidence from the case projects to corroborate literature findings is that the empirical evidence is based on observed phenomena and derives knowledge from actual experience rather than from theory (Arzarello & Olivero, 2006). In accordance with the problem explicating stage of the design science research framework, a detailed understanding of the problems, which were identified through an extensive literature review, can be assimilated by positioning them in the context where the problems practically occur (Johannesson & Perjons, 2014). The images captured from the case projects were used in the following subsections to demonstrate the technical challenges and their impacts on feature extraction.

As revealed from the literature review, the technical challenges lead to difficulties in robust object detection, classification and segmentation using conventional machine learning algorithms. Most of the existing computer vision based indoor construction progress monitoring studies are centred around conventional machine learning algorithms for recognition of indoor as-built elements. These algorithms are dependent on handcrafted or manual feature designing and extraction. Poor visual quality and the

background noise in the indoor site images triggered by the technical challenges constrain robust feature extraction by conventional machine learning algorithms. Therefore, the images collected from indoor sites should be pre-processed to remove the background noise and enhance the visual quality.

Since the progress monitoring scenario selected for this research is the framing, insulation, and drywall installation of indoor walls, edges are the predominant feature that can be used for feature extraction. In addition to edges, texture, colour, and shape can be used concurrently as features with their corresponding machine learning algorithms. However, for the explanatory purposes of the current study, the **Canny edge detection algorithm** (Canny, 1986), which is a prominent traditional edge detection algorithm, was employed on non pre-processed images to demonstrate the impacts caused by technical challenges in terms of visual recognition of the as-built state of these indoor construction elements. The Python code was developed with the guidance of the documentation in OpenCV (2022a) on `cv.Canny` function using OpenCV version 4.6.0 in Jupyter notebook as depicted in Figure 4.1.

```
# Importing the libraries
import numpy as np
import cv2
# Loading the image in full colour
img = cv2.imread('image1.jpg',1)

# Assigning threshold values
hsv = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
res, thresh = cv2.threshold(hsv[:, :, 0], 25, 255, cv2.THRESH_BINARY_INV)
cv2.imshow("Thresh", thresh)

# Providing the lower and upper limit of the threshold of Canny edges
edges = cv2.Canny(img,100,70)
# Displaying the Canny edge detected image
cv2.imshow("Canny", edges)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Figure 4.1: Sample of Python code for Canny edge detection

To implement edge detection using Canny, first the libraries are imported, then the images in their full colour are loaded. Then the Hue Saturation Value (HSV) colour space based thresholding is assigned to determine the pixel values to be extracted. For Canny edge detection, HSV colour space is selected over Red, Green, Blue (RGB) colour space,

because the HSV is more robust for external lighting changes (OpenCV, 2022a). The lower limit of the threshold of the edges and the upper limit of the threshold of the edges are provided to display the Canny edge detected image. From the Python code, which is illustrated in Figure 4.1, it can be discerned that assigning the thresholding values requires manual intervention. The edge detection is sensitive to noise such as occlusions and illumination changes in the image (OpenCV, 2022a). Therefore, when using traditional machine learning algorithms such as Canny edge detector for object detection, the values assigned to detect the feature “edges” must be modified in each image.

The images captured from the case projects were used in the following subsections to demonstrate the technical challenges and their impacts on feature extraction when a traditional machine learning algorithm such as Canny edge detector is used.

4.1.1 Indoor objects related challenges

In heavily cluttered indoor construction sites, detection of the region of interest from the site images is constrained by the occlusions caused by construction equipment and materials. Construction materials and equipment such as stepladders and scaffolding constrain the robust detection of region of interest in images due to the occlusions, they cause. Figure 4.2 illustrates how the region of interest of a drywall has been partially occluded by a stepladder.



Figure 4.2: Construction equipment creating occlusions

In addition to construction material and equipment, movements of construction personnel create cluttered indoor scenes and occlusions in images. Feature extraction of the framing area is constrained by the presence of a stepladder, framing material and movements of construction personnel partially occluding the framing area. This is evidenced in Figure 4.3.



Figure 4.3: Construction personnel and material creating occlusions

In addition to the construction workers, when an occupied building is renovated, building occupants can also accidentally enter the construction site. Therefore, the clutter and occlusions in images caused by human involvement are comparatively higher in indoor sites. The occlusions and clutter caused by construction materials, equipment and human movements create background noise when extracting features from the region of interest using handcrafted algorithms.

Movements in the interior walls cause overlapping images to be captured. When capturing the progress images of drywall installation, the moving of drywall sheets while they are being installed causes clutter in enclosed sites as indoor sites, disrupting the data collection. It is difficult to observe the state transformation clearly with such movements as shown in Figure 4.4. Therefore, many overlapping images must be collected from the same region of interest to present detectable features for the handcrafted algorithms.



Figure 4.4: Movement of the drywall sheets

The attributes of indoor objects such as the shape, texture, edges, and colours play a key role in feature extraction using traditional machine learning algorithms. As illustrated in Figure 4.5, since insulation blankets and drywall sheets have a distinctive texture, when a Canny edge detector is employed, the insulation state and the drywall installation state can be distinguished despite the partial occlusion created by the stepladder.

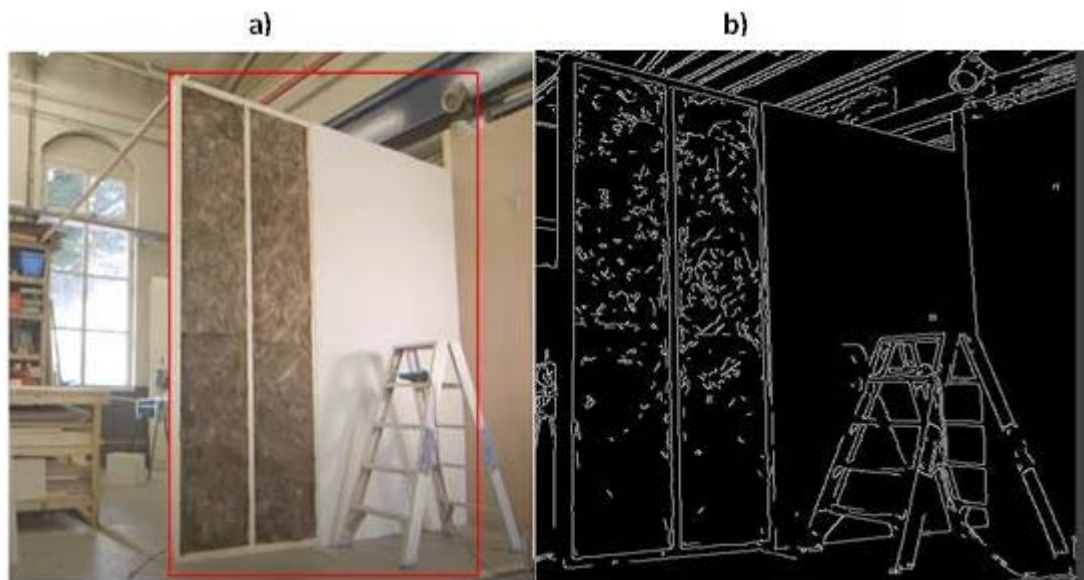


Figure 4.5: a) Original image; b) Canny edge detector extracting insulation and drywall states

Removal of the stepladder from the background of the image would have made the edge detection algorithm extract features more robustly. Although the visual recognition of the drywall surface can be achieved with partial occlusions, segmenting the pixel mask of the drywall area by traditional machine learning algorithms for work-in-progress calculation can be constrained by such occlusions. However, removing occlusions from the images requires highly manual intervention as the nature and number of occlusions in one image may be different from another image. Generally, overlapping images of the same scenario and region of interest are captured to reduce the impacts of occlusions.

4.1.2 Lighting conditions related challenges

The major challenge in indoor sites compared to exterior sites is the lack of natural light which in turn creates low illuminated environments. The image capture resolution tends to be higher in outdoor sites in natural lighting. An example of low illumination in indoor environments is shown in Figure 4.6. This indoor scene was illuminated by an artificial lighting fixture. However, at the time this image was captured, other than the natural lighting entering the indoor location from a small opening in the bottom left corner, no other lighting source was used in this indoor scene.



Figure 4.6: Low illumination in the indoor site

When there is low illumination, feature extraction from the images of low resolution becomes cumbersome. In addition to the low illuminated indoor scenes, it was observed that the natural light entering the indoor site from openings produces backlight and causes shadows as shown in Figure 4.7. These illumination changes constrain feature extraction

by conventional machine learning algorithms as the region of interest is not completely detectable.



Figure 4.7: Backlight caused by natural light entering the site

The artificial lights installed in indoor sites cause noise on the cameras during data collection. The impact of bright artificial light on detecting an insulated wall is shown in Figure 4.8. Extracting the features to detect the region of interest can be hampered by changing indoor lighting due to variations in pixel intensities. Counteracting the effects of non-uniform illumination involves illumination enhancement pre-processing algorithms such as brightness equalisation. Since the brightness exposure of one image is different from another image, adjusting the brightness levels using this type of pre-processing requires heavy manual intervention.



Figure 4.8: Extracting the region of interest affected by a bright artificial lighting

The empirical evidence proves that the handcrafted algorithms relying on extracting features such as edges, colours, and texture from the region of interest to detect the objects are affected by bright indoor artificial lighting fixtures, low illumination, and backlight. The region of interest recognition is intricate when the edges of the studs, which are affected by the backlight, are extracted using Canny edge detection algorithm as shown in Figure 4.9.

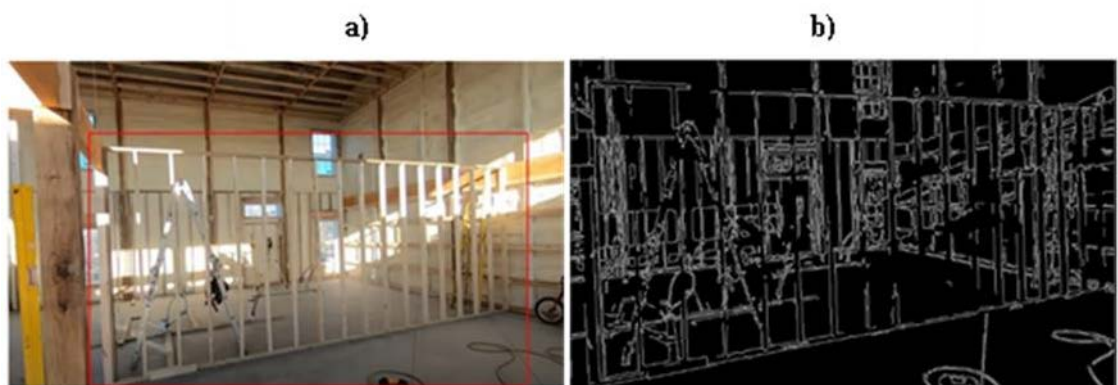


Figure 4.9: a) Original image; b) Subjected to Canny edge detection

The studs affected by the shadows and backlight are missing in the Canny edge detected image in Figure 4.9. Therefore, when handcrafted machine learning algorithms are employed, the illumination intensity threshold values must be adjusted because illumination intensities are used to differentiate objects from the background. As a result, image pre-processing is crucial to visually enhance the features before performing detection if handcrafted algorithms are used.

4.1.3 Camera positioning related challenges

The choice of using fixed cameras over UAVs, UGVs and web cameras for image data collection was justified in section 3.2.2. Employing UAVs or UGVs for data collection would have been impractical in the heavily congested selected indoor sites due to the safety of the workers, potential disturbance to workers and legal regulations on flying drones in Australia. Since there was no stable internet connection at the selected indoor sites, there would have been an additional cost to set up the web camera connection. On the other hand, using fixed time-lapse cameras proved to be more practical during the global COVID pandemic due to the rules for social distancing and minimal contact. Although fixed cameras were used to collect the indoor site images for the current study, fixed cameras have certain challenges.

When using fixed cameras, the major problem is to determine the best vantage point to install the camera. This strategic location depends on the availability of the contractor's resources including scaffolding and indoor site structures such as columns and beams. Striking a balance between the best viewing angle of the region of interest and the availability of the best vantage point in an enclosed indoor space for the camera positioning is challenging. As a result, the camera may not be positioned at the right angle to the region of interest. One example of the angle of the region of interest of framing being slightly inclined due to the location of the camera is shown in Figure 4.10. The camera was attached to scaffolding.



Figure 4.10: The slightly inclined region of interest captured by the camera

Due to the limited field of view of the camera, the image is susceptible to distortion and angle effect. In another example, as evidenced in Figure 4.11, the camera was installed on a beam to capture the region of interest inside the red-coloured boxes. The drywall sheets of the entire enclosed room are not observable inside the field of view of the camera. In a similar situation of capturing the region of interest, the camera was fixed on a column, but the camera location was far from the region of interest of both framed walls as displayed in Figure 4.12.



Figure 4.11: Field of view is affected by camera positioning



Figure 4.12: Camera location is far from the walls

The obstruction to the line of sight of the camera is also caused by part of the safety net as shown in Figure 4.12. In addition to these challenges, the field of view of the camera is obstructed by the movements of construction personnel and temporary construction objects. These obstructions are shown in Figure 4.13. The camera was installed on scaffolding to capture the framing process because it provided the best vantage point.

However, when the site personnel wanted to move the scaffolding, finding another suitable location for the camera installation was difficult.



Figure 4.13: Obstruction to the field of view of the camera

The use of scaffolding to hold the camera also presented the unexpected drawback of construction personnel accidentally knocking over the camera while they were working. Unsupervised cameras may lead to uncertainties and cause the loss of image data of the region of interest, as shown in Figure 4.14. Since time-lapse cameras generate a large number of overlapping images, capturing the region of interest of the progress monitoring scenario was not hindered.



Figure 4.14: Camera being accidentally turned away

Regular check-ups were effective to avoid memory and battery outage, unsupervised camera movements and damage to the cameras in heavily cluttered and occluded indoor sites. To avoid such situations, cameras were checked on a fortnightly basis. Although it required manual intervention, such efforts are also necessary to facilitate the relocation of cameras as the construction activities of the progress monitoring scenario occurred in

different locations of the site. As pointed out in section 3.2.1, because of the COVID lockdown in NSW, the researcher was not able to visit Project 2 until the restrictions were lifted on 11 October 2021. During that time, the batteries of one of the cameras in Project 1 were out of charge. Although the site works of the selected progress monitoring scenario did not commence until 14 October 2021, it is evident that when conducting this type of data collection, if the cameras are left unsupervised for a prolonged period, the risk of running out of battery life is high.

Compared to indoor object and lighting condition related challenges, the camera positioning related challenges are mostly equipment and strategic position based practical limitations. When the images are captured from different perspectives and angles, the images can be manually augmented by cropping and resizing to obtain a clear region of interest. The image augmentation techniques have been presented in a Python code in Figure 3.8 in section 3.2.3. When using machine learning algorithms with handcrafted feature extraction, the visual recognition of the region of interest is constrained if the objects are obstructing the camera's field of view and partially occluding the region of interest of the images. If the image is processed with Canny edge detector without initially being subjected to augmentation, the results would be too complicated to infer as evidenced in Figure 4.15.

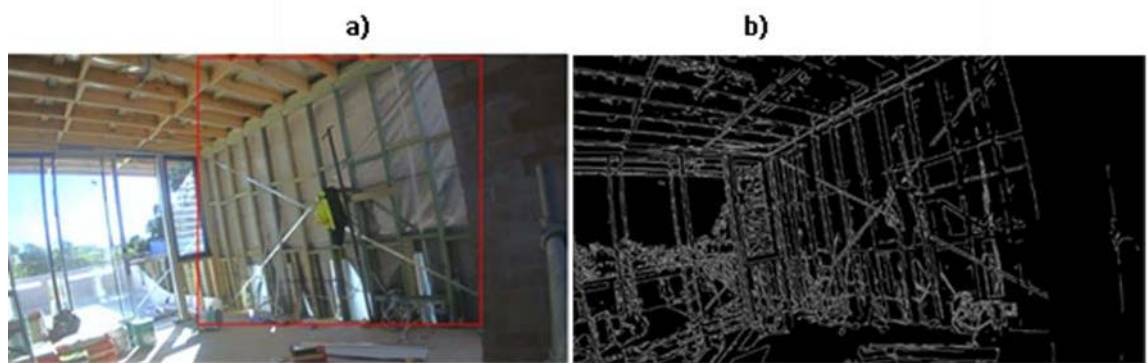


Figure 4.15: a) Original image; b) Original subjected to Canny edge detection

The Canny edge detector algorithm detects all the edges in the image as shown in Figure 4.15. Therefore, it is crucial to localise the region of interest to avoid complications. Figure 4.16 shows how the image is augmented to highlight the region of interest.

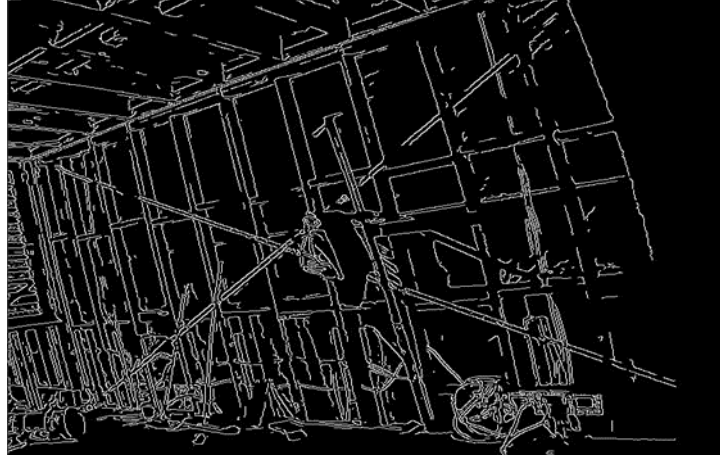


Figure 4.16: Region of interest subjected to Canny edge detection

The resized and cropped region of interest is subjected to Canny edge detection to visually recognise the framing area in the image. Although the studs are detected through edge detection, without correcting the angular and distorted perspective of the image, the edge detected image appears very intricate.

Since the challenges discussed in sections 4.1.1, 4.1.2 and 4.1.3 constrain robust object detection, classification and segmentation of indoor construction elements, the automated visual recognition of the as-built state and the work-in-progress measurement are immensely affected. After determining these challenges, the next section focuses on the application of image pre-processing techniques to reduce the impacts of these challenges.

4.2 Pre-processing techniques to reduce the impacts of the technical challenges

Implementing algorithmic techniques to pre-process the images from the indoor construction sites contributes to robust feature extraction. The pre-processing algorithms may entail different levels of manual intervention depending on the level of corrections required. In the current study, the feature extraction from indoor site images uses a deep learning approach opposed to the traditional machine learning algorithms used for the same tasks by most of the previous studies. While deep learning algorithms are proven to be accurate and efficient for feature extraction compared to conventional machine learning algorithms, pre-processing with a low level of manual intervention may be required depending on the visual quality of the image. This is significant, when capturing

overlapping images using time-lapse cameras and augmenting images to extract the region of interest are not sufficient. The current study presents two pre-processing algorithmic techniques with a minimum amount of manual intervention. A perspective transformation code for correcting the angular view of the region of interest affected by the distortions and a low illumination enhancement code for enhancing the illumination of the images captured under dark conditions are presented.

4.2.1 *Perspective transformation in image processing*

Because of the positioning of the camera, the region of interest of the site images are affected by distortions thus causing an inclined and angular view. Using the **cv2.getPerspectiveTransform** and then **cv2.warpPerspective** functions of OpenCV (OpenCV, 2022b), the original perspective can be modified to obtain an accurate region of interest. A Python code was developed for perspective transformation using OpenCV version 4.6.0 in Jupyter notebook. This code is presented in Figure 4.17.

```
#Importing the Libraries
import cv2
import numpy as np

#Loading the original image
img = cv2.imread("images/img10.jpg")

#Pixel value points of the original image
input_points = np.float32([[434, 212], [904, 1], [497, 589], [1019, 671]])

#Desired pixel value points in the output image
converted_points = np.float32([[0, 0], [1000, 0], [0, 700], [1000, 700]])

#Perspective transformation
matrix = cv2.getPerspectiveTransform(input_points, converted_points)
img_output = cv2.warpPerspective(img, matrix, (1000, 700))

#Displaying the original image and perspective transformed image windows
cv2.imshow("Original", img)
cv2.imshow("Perspective transformation", img_output)
cv2.waitKey(0)
```

Figure 4.17: Python code for perspective transformation

In accordance with the perspective transformation procedure, first the libraries are imported. Then the image in which the perspective needs to be modified is loaded. In the next step, the pixel points of four corners of the region of interest of the loaded image are determined. These four points are the top left corner point (x_2, y_2), top right corner point (x_3, y_2), bottom left corner (x_1, y_1) point and bottom right corner point (x_2, y_1). If the pixel

coordinates of the four corners in the original image are defined as P_0 , it can be presented as in Equation 4.1.

$$P_0 = [(x_2, y_2), (x_3, y_2), (x_1, y_1), (x_2, y_1)] \quad \text{Equation 4.1}$$

After determining the original pixel coordinates of the four corners of the region of interest, the desired pixel coordinates of the four corners in the output image are defined as P_n as in Equation 4.2.

$$P_n = [(0, 0), (x_n, 0), (0, y_n), (x_n, y_n)] \quad \text{Equation 4.2}$$

The resultant image is warped based on the pixel width (x_n) and the pixel height (y_n) values, which are decided in this step. The original image is displayed with the perspective transformed image to recognise the differences. The region of interest in the original image of Figure 4.18a had distortion and an angular effect due to camera positioning. After correcting the image through perspective transformation, the region of interest of the framing was obtained as shown in Figure 4.18b.



Figure 4.18: Perspective transformation: a) Original image; b) Corrected image

Although the perspective transformation technique is effective for correcting the perspective of the region of interest of the image, determining the pixel points in the region of interest in the original image and assigning the desired pixel points for the warped image requires manual intervention.

4.2.2 Low illumination enhancement in image processing

The low illumination problem in indoor sites contributes to creating very dark images from which the feature extraction is challenging unless the images are corrected with lighting enhancements. Many conventional image processing techniques that are used to enhance images captured in low illuminated environments are heavily manually

intervened to adjust the exposure levels and to establish the threshold values (Shi et al., 2018). The **dual channel (dark and bright) based method** for low illumination image enhancement proposed by Shi et al. (2018) is an effective image processing technique to remove the block effects in the dark regions of the image and enhance the illumination of the region of interest. This technique was used in the current study. When a low illuminated image I is input, its image exposure model can be represented as in Equation 4.3.

$$I(x) = J(x)t(x) + A(1 - t(x)) \quad \text{Equation 4.3}$$

where $J(x)$ is the enhanced or the rectified image, $I(x)$ is the low illuminated image, A is the atmospheric light, and $t(x)$ is the corrected transmission map. The purpose is to improve the low illuminated image I and obtain the enhanced image J as in Equation 4.4.

$$J(x) = \frac{I(x) - A}{t(x)} + A \quad \text{Equation 4.4}$$

From Equation 4.4, it can be discerned that if $t(x)$ and A are identified, the enhanced image J can be obtained. The technique to transform the low illuminated image to enhanced image is through the dual channel (dark and bright) based method (Shi et al., 2018). The bright and dark channel prior represents the maximum and minimum intensity of pixels in a local patch respectively.

$$I^{dark}(x) = \min_{c \in \{r, g, b\}} (\min_{y \in \Omega(x)} (I^c(y))) \quad \text{Equation 4.5}$$

$$I^{bright}(x) = \max_{c \in \{r, g, b\}} (\max_{y \in \Omega(x)} (I^c(y))) \quad \text{Equation 4.6}$$

I^c is a colour channel of I and $\Omega(x)$ is a local patch centred at x . Corresponding to these values, y is a pixel in the local path $\Omega(x)$. The process to execute the dual channel (dark and bright) based method was implemented using OpenCV version 4.6.0 in Jupyter notebook. The code can be explained in two parts as shown in Figure 4.19 and Figure 4.20 respectively.

```

# Importing the libraries
import cv2
import numpy as np

# Loading the image and a copy of the image
im = cv2.imread("Image1.jpg")
orig = im.copy()

# Defining the illumination channel
#storing the image dimensions in M and N variables
def get_illumination_channel(I, w):
    M, N, _ = I.shape
# Applying padding of half of the kernel size
    padded = np.pad(I, ((int(w/2), int(w/2)), (int(w/2), int(w/2)), (0, 0)),
                    'edge')
    darkch = np.zeros((M, N))
    brightch = np.zeros((M, N))

    for i, j in np.ndindex(darkch.shape):
        darkch[i, j] = np.min(padded[i:i + w, j:j + w, :])
        brightch[i, j] = np.max(padded[i:i + w, j:j + w, :])

    return darkch, brightch

# Reshaping, flattening and sorting the image array
def get_atmosphere(I, brightch, p=0.1):
    M, N = brightch.shape
    flatI = I.reshape(M*N, 3)
    flatbright = brightch.ravel()

    searchidx = (-flatbright).argsort()[int(M*N*p):]
    A = np.mean(flatI.take(searchidx, axis=0), dtype=np.float64, axis=0)
    return A

```

Figure 4.19: Python code for low light enhancement: Part I

The first part of the code is shown in Figure 4.19. First, the libraries are imported. The low illuminated image and a copy of the image are also loaded. In the next step, a function is written to load the illumination channel. The image dimensions are stored in the variables named M and N. To ensure the dimensions remain the same, a padding of half the kernel size is used for the image. The dual channels, i.e. dark and bright channels, are acquired by initialising a matrix. The dark channel is acquired using “np.min” to obtain the lowest pixel value whereas the bright channel is acquired by using “np.max” to obtain the highest pixel value. The next step is to arrange the image array. The image array is

reshaped, flattened, and sorted according to the maximum intensity. The second part of the code is presented in Figure 4.20.

```
# Finding the initial transmission map
def get_initial_transmission(A, brightch):
    A_c = np.max(A)
    init_t = (brightch-A_c)/(1.-A_c)
    return (init_t - np.min(init_t))/(np.max(init_t) - np.min(init_t))

# Estimating the corrected transmission map
def get_corrected_transmission(I, A, darkch, brightch, init_t, alpha, omega, w):
    im3 = np.empty(I.shape, I.dtype);
    for ind in range(0, 3):
        im3[:, :, ind] = I[:, :, ind] / A[ind]
    dark_c, _ = get_illumination_channel(im3, w)
    dark_t = 1 - omega*dark_c
    corrected_t = init_t
    diffch = brightch - darkch

    for i in range(diffch.shape[0]):
        for j in range(diffch.shape[1]):
            if(diffch[i, j] < alpha):
                corrected_t[i, j] = dark_t[i, j] * init_t[i, j]

    return np.abs(corrected_t)

# Calculating the normalised image
def get_final_image(I, A, refined_t, tmin):
    refined_t_broadcasted = np.broadcast_to(refined_t[:, :, None],
        (refined_t.shape[0], refined_t.shape[1], 3))
    J = (I-A) / (np.where(refined_t_broadcasted < tmin,
        tmin, refined_t_broadcasted)) + A
    return (J - np.min(J))/(np.max(J) - np.min(J))

# Displaying the corrected image
cv2.imshow('original', orig)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

Figure 4.20: Python code for low light enhancement: Part II

In the second part of the code, finding and initialising the transmission map is prioritised. The transmission map illustrates the portion of the light that is not scattered. This portion of light usually reaches the camera. In the next step, the parameter ω , which is applied to correct the initial transmission map, is usually set to 0.75. After correcting the potential erroneous transmission estimates in the dark and bright channels, the corrected transmission map is executed as the initial transmission map. This corrected transmission

map is crucial to calculate the normalised image. In the final step, the corrected image is displayed along with the originally loaded low illuminated image. The resultant image is comparatively illuminated compared to the original dark image. An example of the low illumination enhancement of indoor site images is displayed in Figure 4.21.

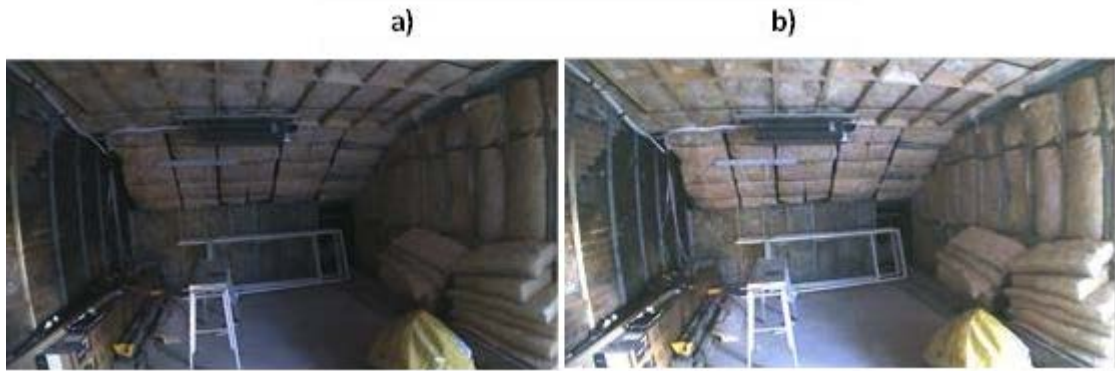


Figure 4.21: Low illumination enhancement: a) Original image; b) Enhanced image

The dual channel (dark and bright) based method consists of many underlying mathematical operations. As a result, the implementation of this technique to enhance the illumination of images captured in low lighting conditions requires less manual intervention in terms of adjusting the exposure levels and establishing the threshold values.

Pre-processing with a low level of manual intervention as shown in the perspective transformation and low illumination enhancement algorithms may be required depending on the visual quality of the indoor site images.

4.3 Discussion

As encapsulated from both literature and case projects, these technical challenges are presented in Table 4.1.

Table 4.1: Technical challenges for as-built indoor elements visual recognition

Indoor objects related	Lighting conditions related	Camera positioning related
Inherent characteristics (shape, texture, edges, size) of indoor objects constrain feature extraction.	Indoor sites have low illuminated environments compared to outdoor sites which use natural lighting.	Fixed cameras get hindered by the limited data capturing range and viewing angle.
Movements and changes of interior elements (walls, indoor objects) affect the state transformation recognition.	Artificial light sources, backlights and shadows constrain region of interest recognition in indoor images.	Unsupervised cameras and their movements lead to loss of image data capture.
Occlusions may occur in as-built scenes due to temporary equipment and materials.	Changes in pixel intensities in indoor images caused by both natural and artificial lighting constrain feature extraction.	Relocation of cameras with construction progression, requires manual intervention.
Cluttered indoor scenes are caused by human involvement.		To generate complete as-built models for indoor sites, a large set of overlapping images is required due to the limited visibilities and field of view of cameras.
		When fixed cameras are used indoors, the line of sight is obstructed as walls are being erected.
		Using UAVs or UGVs is impractical in the heavily congested indoor sites due to the safety of the workers, potential disturbance to workers and legal regulations on flying drones.
		If the cameras are unsupervised for a long period, battery outage and damage to the camera can be expected.

It was evident that compared to exterior sites, the inherent characteristics of indoor objects constrain object recognition through feature extraction by handcrafted machine learning algorithms such as Canny edge detector. This is a limitation in the majority of the existing computer vision based indoor construction element recognition studies such as those conducted by Hamledari and McCabe (2016); Hamledari et al. (2018); Kropp et al. (2014) and Kropp et al. (2018). This is because they have used handcrafted machine learning algorithms to extract shape, colours, edges, and texture and thereby determine the as-built

state of indoor construction elements. In their study using a deep learning model instead of conventional machine learning algorithms for recognition of indoor elements, Ying and Lee (2019) pointed out that deep learning algorithm's ability to learn features from images by training thousands of images contributes to automated feature extraction. Studies such as Hamledari and McCabe (2016); Kropp et al. (2012) and Kropp et al. (2014) highlighted the importance of removing background noise through pre-processing algorithms such as frame differencing (Kartika & Mohamed, 2011). However, the degree of occlusions in the background and clutter in an indoor site varies from site to site. Therefore, applying pre-processing to remove clutter and occlusions related background noise requires heavy manual intervention. As confirmed from the empirical evidence of this study, using time-lapse photography enabled generating many overlapping images, and thus capturing the region of interest of the progress monitoring scenario for feature extraction was not hindered. In addition, since deep learning algorithms do not depend on a single image to learn features of a certain region of interest (O'Mahony et al., 2019), providing many overlapping images could resolve this problem. Therefore, in this study, a pre-processing algorithm to reduce the background noise created by occlusions and clutter is not presented.

The empirical evidence proved that the low illumination in the indoor sites is a unique challenge for automated visual recognition of indoor elements compared to object recognition in an exterior site with natural light. Additionally, the inconsistent illumination intensities in indoor site images and the different exposure as a result of artificial lighting, backlight and shadows constrain feature extraction. Hamledari and McCabe (2016) argued that the visual recognition algorithms must be adjusted according to the lighting conditions. In their study of using handcrafted machine learning algorithms for automated visual recognition of indoor elements, Kropp et al. (2013) used a contrast stretching pre-processing algorithm to normalise illumination which requires manual intervention to a greater degree (Kropp et al., 2013). Although W. Wei et al. (2022) used a deep learning model, which is capable of feature learning from the images, W. Wei et al. (2022) also applied a brightness equalisation pre-processing algorithm to equalise illumination intensities. However, their method requires heavy manual intervention depending on the degree of illumination intensity in indoor environments. To reduce the manual intervention, the low illumination enhancement pre-processing algorithmic

solution presented in the current study only focuses on increasing the illumination of the images captured in low lit indoors.

The empirical evidence showed that digital cameras that use time-lapse photography mounted at a fixed location in the site are an effective means of capturing data from enclosed indoor sites, compared to other camera equipment and mounting techniques. In Hamledari and McCabe (2016) and Hamledari, McCabe and Davari (2017), UAVs were used to capture indoor site images. It was considered impractical in the current study to use UAVs or UGVs in the heavily congested indoor sites due to the safety of the workers, potential disturbance to workers and legal regulations on flying drones. The studies of Ying and Lee (2019) and W. Wei et al. (2022) used a handheld camera to capture the indoor images and this method required heavy manual intervention. The current study acknowledged the main challenge related to fixed cameras is determining the strategic location to install the cameras to capture the region of interest. If the cameras are not strategically positioned, the region of interest of the image is susceptible to distorted perspective creating background noise. Therefore, a perspective transformation pre-processing algorithmic solution for correcting the angular view of the region of interest affected by the distortions was presented.

As shown in the evidence from previous studies and the empirical evidence gathered from the case projects, compared to the traditional machine learning algorithms, applying deep learning algorithms is proven to be comparatively accurate and efficient in terms of feature extraction. Nevertheless, depending on the visual quality of the indoor site images, pre-processing with a low level of manual intervention may be required to facilitate robust feature extraction.

4.4 Chapter summary

The empirical evidence gathered from the case projects contributed to corroborating and modifying the literature findings in section 2.5.1. These technical challenges are challenges related to indoor elements, lighting conditions and camera positioning. Their impacts for feature extraction, as demonstrated by using the machine learning algorithm, i.e. Canny edge detector, proved that conventional algorithms are not robust for visual recognition of indoor elements and demonstrated the impacts of technical challenges for computer vision based as-built indoor elements visual recognition. When the visual quality of the indoor images is poor, the two pre-processing techniques of perspective

transformation and low illumination enhancement as presented in this study can be used to facilitate robust feature extraction. This chapter presented the results to fulfil Objective 1 of the research, which is to investigate the technical challenges and their impacts for visual recognition of indoor as-built elements.

5 Results and discussion: Part II

After investigating how the technical challenges in indoor construction environments affect the visual recognition of as-built elements in Chapter 4, this chapter explores how deep learning can be used as a solution. This chapter has two main parts. Section 5.1 presents how the Mask R-CNN model can be optimised by transfer learning and fine-tuning. This optimised deep learning model is used for automated visual recognition of indoor as-built elements. Section 5.2 focuses on the work completion calculation of indoor elements using the post-processing equations applied to the optimised Mask R-CNN model. The post-processed Mask R-CNN model can extract the areas of the segmented pixel masks corresponding to the as-built states detected. This pixel information is then used to calculate the work completion percentages of indoor elements. The chapter concludes with a discussion in section 5.3 and a summary of the findings in section 5.4.

5.1 Automated visual recognition of indoor as-built elements

The Mask R-CNN model, originally introduced by K. He et al. (2017), was considered as the base model, to which the transfer learning was applied and certain hyperparameters of the base model were fine-tuned. The transfer learning and hyperparameter fine-tuning strategy resolves the onerous process of building the deep learning model from scratch. Compared to the base model, the optimised model was improved in terms of the detection, classification and segmentation accuracy for the custom dataset comprised of framing, insulation, and drywall installation images of indoor walls. This optimised model was trained on Google Colab to reduce the high computational resource requirements associated with deep learning models. Using Colab Pro is a cheaper option, with the resources provided by Google compared to employing a physical computer, which requires high performing GPU, RAM and CPU with computing platforms and libraries installed manually. Due to the improved detection, classification and segmentation accuracy, the model is capable of minimising the impacts of technical challenges related to indoor objects, lighting and camera positioning in indoor construction environments.

5.1.1 The workflow of modifying Mask R-CNN and training on Colab

The steps followed in modifying and training the Mask R-CNN model are explained in the following subsections. The Mask R-CNN model was trained using the “Detectron2”

library with the “PyTorch” framework. The technical algorithm representing the workflow of modifying the Mask R-CNN model is shown in Figure 5.1.

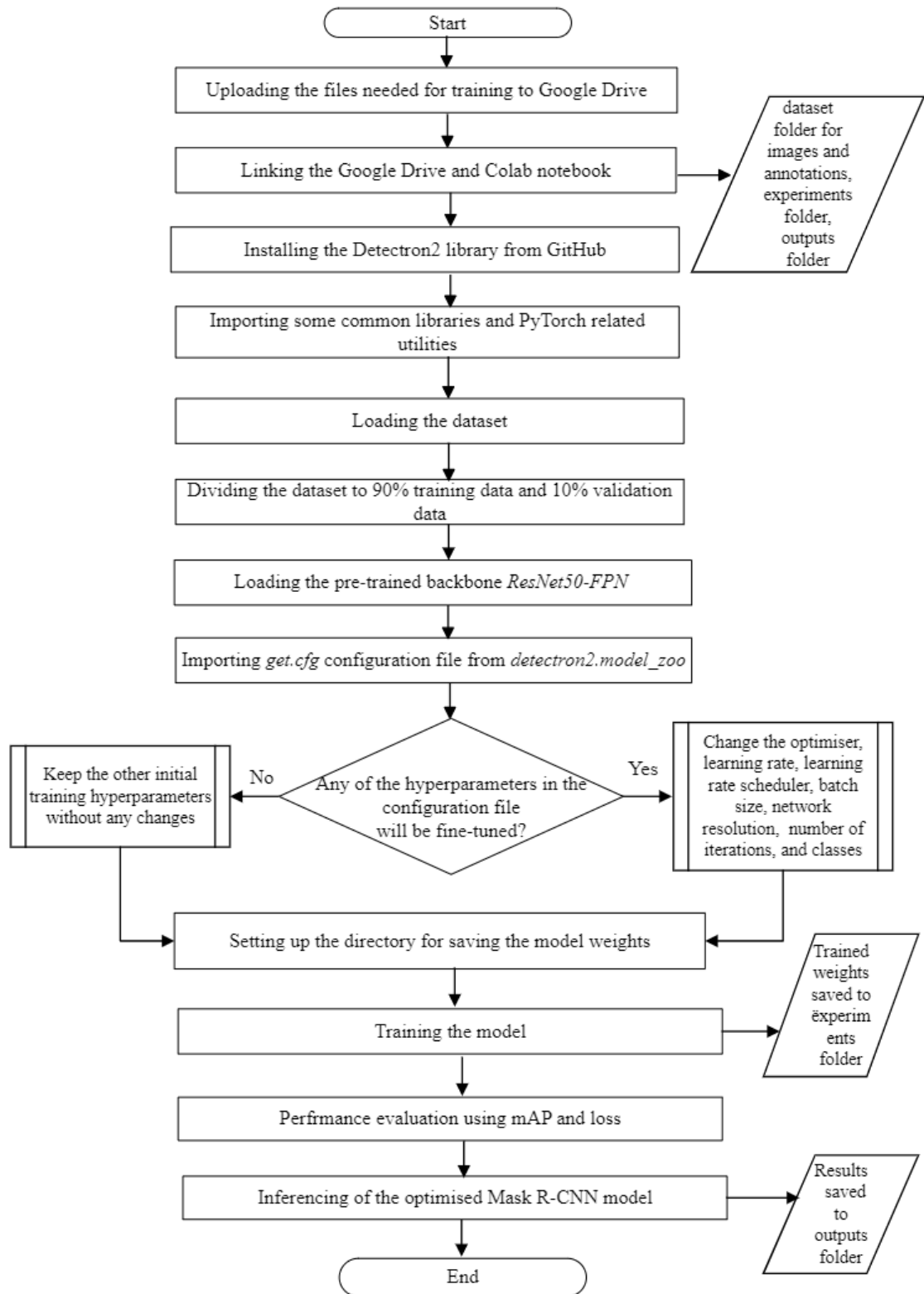


Figure 5.1: The technical algorithm

The technical algorithm is further elaborated in the following subsections to understand the building and training process of the optimised model.

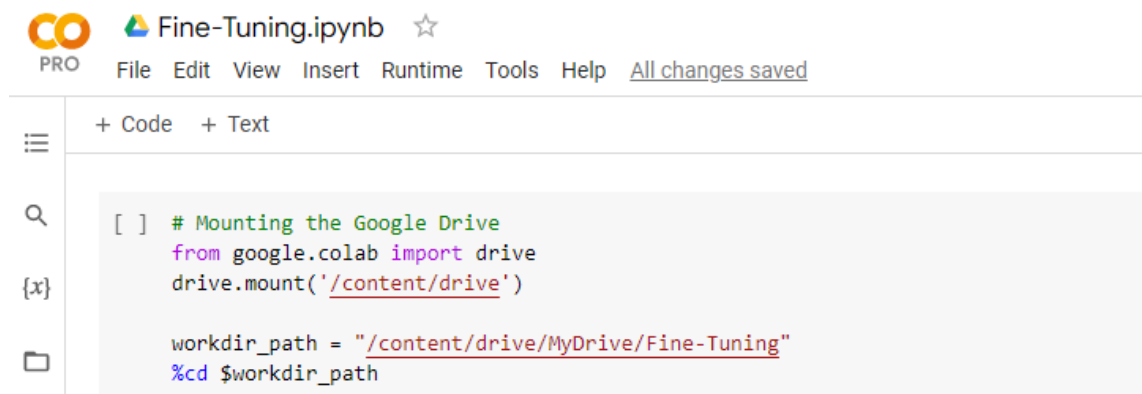
Step 1: Uploading the files needed for training to Google Drive

Google Drive is the storage location for Colab. Therefore, before executing training on Colab, the supporting files for training should be uploaded to Google Drive. A folder named “*fine-tuning*” was created in Google Drive under “*My Drive*” location. The following folders were uploaded to this “*fine-tuning*” folder. The naming convention was tailored to reflect the purpose of each file:

- “datasets”: The folder contains the image dataset subfolder and the corresponding COCO json annotations to indicate the coordinates.
- “experiments”: The subfolder saves the weights of the optimised Mask R-CNN model trained on the image dataset.
- “output”: The folder saves the segmented images.

Step 2: Linking Google Drive and Colab notebook

The Colab notebook “*Fine-Tuning.ipynb*” was created from the same Google account linked to Google Drive and was used to execute the Python code for training. This Colab notebook was saved to Google Drive. The “*mount drive*” command was executed to link the “*Fine-Tuning*” folder to the Colab notebook as evidenced in the code shown in Figure 5.2.

The image shows a screenshot of a Google Colab notebook interface. At the top, the Colab logo and 'PRO' are visible, followed by the notebook title 'Fine-Tuning.ipynb' and a star icon. Below the title is a menu bar with options: File, Edit, View, Insert, Runtime, Tools, Help, and a link 'All changes saved'. The main area of the notebook shows a code cell with the following Python code:

```
[ ] # Mounting the Google Drive
from google.colab import drive
drive.mount('/content/drive')

workdir_path = "/content/drive/MyDrive/Fine-Tuning"
%cd $workdir_path
```

Figure 5.2: Mounting Google Drive to Colab

For this study, Colab was connected to a hosted runtime and the runtime was set to GPU and high RAM capacity. At the time of executing this deep learning model, the Colab Pro version offered NVIDIA Tesla P100 GPU of 16 GB with CUDA version 11.2, RAM capacity of 26 GB and Intel® Xeon® CPU of 2.2 GHz in the virtual machine environment. Google Drive has a capacity of 100 GB space, and the hard disk space

provided by Colab is 130 GB. The virtual machine runtime per one connection is up to 24 hours. The specifications of the Colab virtual machine are shown in Figure 5.3.

```
# Check the GPU
!nvidia-smi -L

GPU 0: Tesla P100-PCIE-16GB (UUID: GPU-059e1208-d634-6a90-51bd-ba7482c23b2a)

#Check the CUDA version with GPU
!nvidia-smi
```

Sun Aug 14 07:12:11 2022

NVIDIA-SMI 460.32.03			Driver Version: 460.32.03			CUDA Version: 11.2			

GPU	Name	Persistence-M	Bus-Id	Disp.A	Volatile	Uncorr. ECC			
Fan	Temp	Perf	Pwr:Usage/Cap	Memory-Usage	GPU-Util	Compute M.	MIG M.		
=====									

```
# Check the CPU
!lscpu |grep 'Model name'

Model name:          Intel(R) Xeon(R) CPU @ 2.20GHz

# Check the RAM
!free -h --si | awk '/Mem:/{print $2}'

26G

# Check the hard disk space4
!df -h / | awk '{print $4}'

Avail
130G
```

Figure 5.3: The specifications of the Colab virtual machine

Step 3: Installing the Detectron2 library from GitHub

The custom dataset was trained using the PyTorch framework. “Detectron2 by Facebook Artificial Intelligence Research GitHub repository” was installed in this step. The command used is shown in Figure 5.4.

```
# Installing Detectron2
!python -m pip install pyyaml==5.1
!python -m pip install 'git+https://github.com/facebookresearch/detectron2.git'
```

Figure 5.4: Installing Detectron2

Detectron version 2 is the library which uses the PyTorch framework upon which the Mask R-CNN base model is built.

Step 4: Importing some common libraries and Detectron2 related utilities

Some libraries and Detectron2 related utilities, which are commonly used in PyTorch were imported, and configured in this step as illustrated in the code in Figure 5.5.

```
# import some common libraries
import torch
import time
import datetime
import logging

import numpy as np
import os, json, cv2, random
from google.colab.patches import cv2_imshow

# Setup detectron2 logger
import detectron2
from detectron2.utils.logger import setup_logger
setup_logger()

# import some common detectron2 utilities
from detectron2 import model_zoo
from detectron2.engine import DefaultPredictor
from detectron2.config import get_cfg
from detectron2.utils.visualizer import Visualizer
from detectron2.data import MetadataCatalog, DatasetCatalog
from detectron2.data.datasets import register_coco_instances

from detectron2.engine.hooks import HookBase
from detectron2.evaluation import inference_context
from detectron2.utils.logger import log_every_n_seconds
from detectron2.data import DatasetMapper, build_detection_test_loader
import detectron2.utils.comm as comm

from detectron2.evaluation import COCOEvaluator
from detectron2.engine import DefaultTrainer
```

Figure 5.5: Importing some common libraries and PyTorch related utilities

The version check of the crucial libraries such as PyTorch and OpenCV revealed that PyTorch version 1.21.1 and CV2 version 4.6.0 were used in the Python version 3.7.13 on Colab.

Step 5: Loading the dataset and separating the training and validation datasets

The dataset of 2,000 images with the corresponding annotations was loaded from the “dataset” folder in Google Drive. The paths to the training and validation images with their corresponding annotations were configured as evidenced in Figure 5.6.

```
# Path to training and validation datasets in Google Drive
train_images_path = "./datasets/train/images"
valid_images_path = "./datasets/val/images"
train_annotations_path = "./datasets/train/coco_annotations.json"
valid_annotations_path = "./datasets/val/coco_annotations.json"

# Generate this train and validation json files separately
for dataset_name in ["custom_train_data", "custom_valid_data"]:
    if dataset_name in DatasetCatalog.list():
        DatasetCatalog.remove(dataset_name)

register_coco_instances("custom_train_data", {}, train_annotations_path,
                       train_images_path)
register_coco_instances("custom_valid_data", {}, valid_annotations_path,
                       valid_images_path)

train_metadata = MetadataCatalog.get("custom_train_data")
valid_metadata = MetadataCatalog.get("custom_valid_data")

train_datadict = DatasetCatalog.get("custom_train_data")
valid_datadict = DatasetCatalog.get("custom_valid_data")
```

Figure 5.6: Loading the dataset

The ratio between the training dataset and the validation dataset is 9:1, such that 90% of the dataset (1,800 images) were used for training and 10% of the images (200 images) were used for validation.

Step 6: Loading the pre-trained backbone and fine-tuning the hyperparameters

Because of the smaller dataset and the faster training potential, the ResNet50-FPN model was loaded as the pre-trained backbone. Detectron2 offers a default configuration, including many hyperparameters. To customise the default configuration and optimise the base Mask R-CNN model, *get_cfg* was imported. The configuration file was obtained from the *detectron2.model_zoo*, which is a collection of functions to create common deep learning model architectures using Detectron2. As discussed in section 3.3.2, employing the configuration files from freely available customisable deep learning models such as

“model zoo” (Feurer & Hutter, 2019) is a common method for transfer learning and hyperparameter fine-tuning. It is mostly suitable for models with a smaller dataset.

Section 3.3.2 highlighted that there are currently no simple and easy ways to fine-tune the hyperparameters. Programmers generally conduct many experiments to determine the optimum set of hyperparameters for a given dataset (Smith, 2018). The code used for this purpose is presented in Figure 5.7. The base model with a learning rate of 0.02 with the default SGD optimiser and the learning rate scheduler MultiStepLR (K. He et al., 2017) was initially trained with the hyperparameters shown in Table 5.1. According to PyTorch documentation on the implementation of MultiStepLR, this learning rate scheduler is capable of decaying the learning rate of each parameter group by gamma at the pre-set epoch. This type of decay occurs concurrently with other adjustments to the learning rate.

```
# Load the pre-trained model and fine-tune hyperparameters
pretrained_model_config = "COCO-InstanceSegmentation/mask_rcnn_R_50_FPN_3x.yaml"

cfg = get_cfg()

cfg.merge_from_file(model_zoo.get_config_file(pretrained_model_config))
cfg.DATASETS.TRAIN = ("custom_train_data",)
cfg.DATASETS.TEST = ("custom_valid_data",)
cfg.DATALOADER.NUM_WORKERS = 2
cfg.MODEL.WEIGHTS = model_zoo.get_checkpoint_url(pretrained_model_config)
cfg.SOLVER.IMS_PER_BATCH = 2
cfg.SOLVER.LR_SCHEDULER_NAME = "WarmupCosineLR"
cfg.SOLVER.BASE_LR = 0.00025
cfg.SOLVER.MAX_ITER = 2000
cfg.SOLVER.STEPS = []
cfg.TEST.EVAL_PERIOD = 20
cfg.MODEL.ROI_HEADS.BATCH_SIZE_PER_IMAGE = 128
cfg.MODEL.ROI_HEADS.NUM_CLASSES = 3
```

Figure 5.7: Loading the backbone and fine-tuning the hyperparameters

Except for the **learning rate**, **learning rate scheduler** and **optimiser**, which are directly associated with the optimisation of the deep learning model, other hyperparameters imported using `get_cfg` are affiliated with the efficiency of training of a smaller dataset. Table 5.1 explains the hyperparameters that affect the efficiency of training of the model.

Table 5.1: Customising certain hyperparameters

Hyperparameter	Command	Value
Number of images per batch based on the GPU	cfg.SOLVER.IMS_PER_BATCH	2
Size or the pixel resolution of the images	cfg.INPUT.MIN_SIZE_TRAIN	800x800
Iterations	cfg.SOLVER.MAX_ITER	2,000
Batch size per image	cfg.MODEL.ROI_HEADS.BATCH_SIZE_PER_IMAGE	128
Number of classes to be detected (framed, insulated, drywall_installed)	cfg.MODEL.ROI_HEADS.NUM_CLASSES	3

To determine the optimum performance of the Mask R-CNN model, after experimentation, the hyperparameters explained in Table 5.1 were used with different experimental learning rates and learning rate schedulers with Adam optimiser.

Step 7: Setting up the directory for saving the model weights

The path to save the trained weights from the experiments was set up in the experiments folder in Google Drive. The code snippet related to this step is shown in Figure 5.8.

```
# output_directory
output_dir = "./experiments/"
cfg.OUTPUT_DIR = output_dir
os.makedirs(cfg.OUTPUT_DIR, exist_ok=True)
```

Figure 5.8: Setting up the directory to save the trained weights

Step 8: Training the model

The trainer function was executed to train the Mask R-CNN model. The command is illustrated in Figure 5.9.

```
# Training the model
trainer = MyTrainer(cfg)
trainer.resume_or_load(resume=False)
trainer.train()
```

Figure 5.9: Training the model

Depending on the learning rate, the weights generated from each training were saved to the experiments folder in Google Drive. The training time for a dataset of 2,000 images, in accordance with the hyperparameters presented in Table 5.1, was approximately 3.5 hours.

5.1.2 Performance evaluation of the optimised Mask R-CNN model

The metrics, which were discussed in section 3.3.4, were used to assess the performance of the optimised Mask R-CNN model. Accordingly, the mAP and the loss were the metrics used. The base model to which transfer learning was applied has a learning rate of 0.02. As mentioned in Step 6, many experiments were conducted to find which learning rate provides the best performance with the Adam optimiser, compared to the learning rate of the base model with the SGD optimiser. The learning rate scheduler was also changed from MultiStepLR to cosine annealing (CosineLR). According to the PyTorch documentation, unlike MultiStepLR, the cosine annealing has the effect of rapidly decreasing the learning rate to a minimum value before being increased rapidly again. In Step 7, the trained weights of each of the experiments were saved to Google Drive. The best experiments in relation to mAP and loss values were chosen to determine which learning rate with the Adam optimiser is considered to be the optimised Mask R-CNN model.

The base model (SGD + MultiStepLR+ LR 0.02) by K. He et al. (2017) reported a mAP of 86.72% for the custom dataset (2,000 annotated indoor images on framing, insulation, and drywall installation of indoor walls) used in this study. The learning rate was further reduced to determine which learning rate contributes to a higher mAP. Out of the experiments performed, the learning rate 0.00025 produced a mAP of 86.05% when the SGD optimiser was used with CosineLR (SGD + CosineLR + 0.00025). The optimiser was changed to Adam to experiment with the learning rate 0.00025 and CosineLR scheduler. This experiment reported a mAP of 88.02% (Adam + CosineLR + 0.00025). The comparison of these learning rates with the base model's learning rate is shown in Figure 5.10.

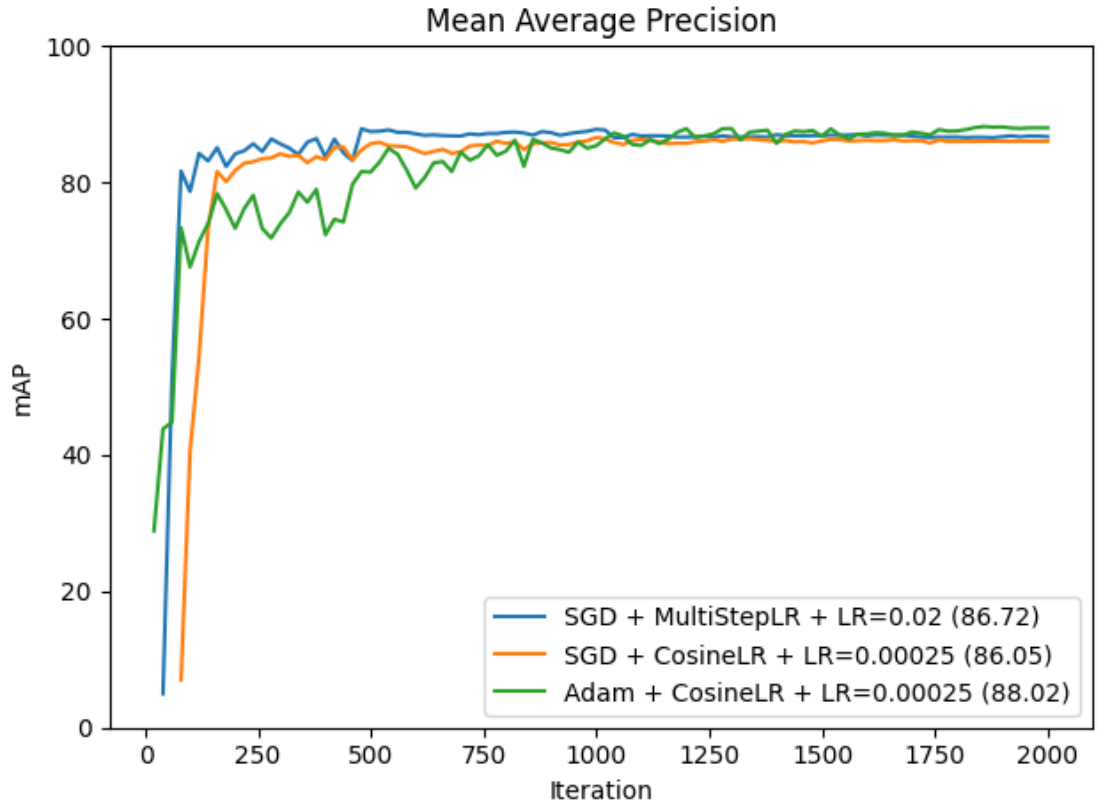


Figure 5.10: Comparison of mAPs with different learning rates

As displayed in Figure 5.10, the model with the learning rate 0.00025 with Adam optimiser and CosineLR has the best mAP of 88.02%, compared to the base model mAP of 86.02%. The optimised model is 1.5% more accurate than the base model in terms of the mAP.

As discussed in section 3.3.4, the training loss and the validation loss were also evaluated to determine the object recognition accuracy of the model. The base model (SGD + MultiStepLR+ LR 0.02) indicated a training loss of 0.21 and a validation loss of 0.21. The model (SGD + CosineLR + 0.00025) hyperparameters showed a training loss of 0.21 and a validation loss of 0.22. The model which used Adam optimiser with the learning rate of 0.00025, scheduled with CosineLR, proved to have the least training and validation losses. The training loss of (Adam + CosineLR + 0.00025) is 0.19 and the validation loss is 0.20. The graph for the training loss is shown in Figure 5.11 and the graph for validation loss is illustrated in Figure 5.12.

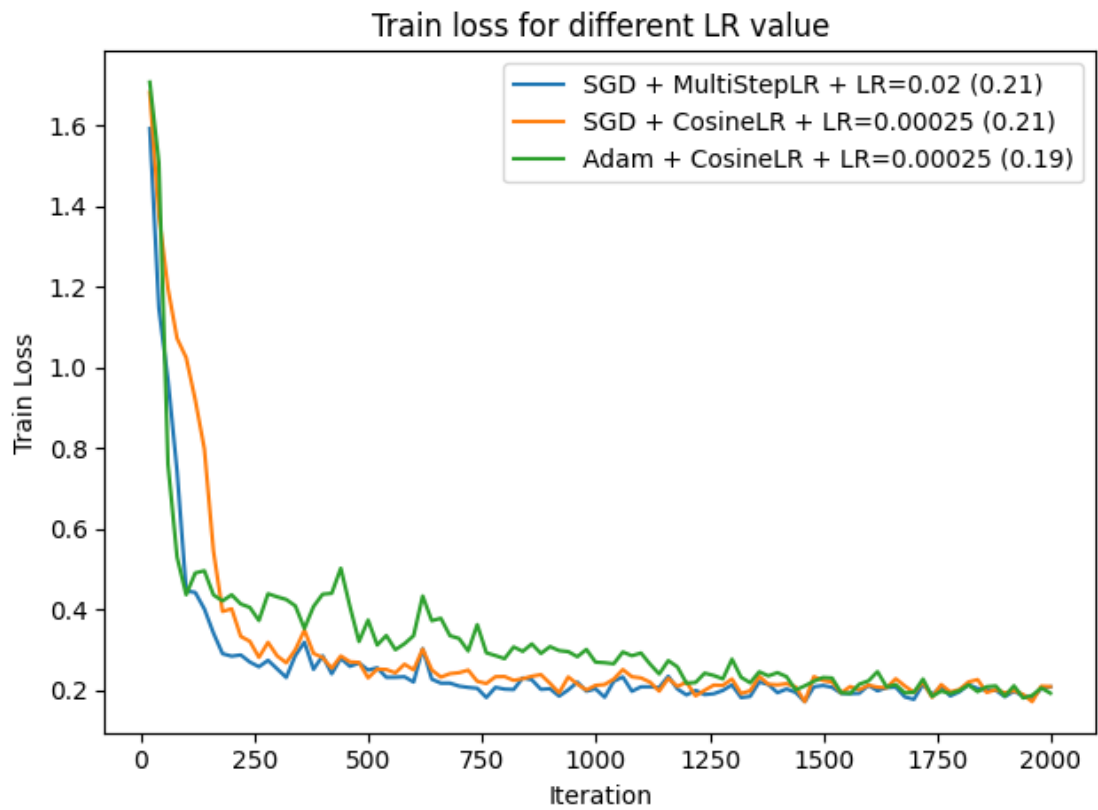


Figure 5.11: Comparison of training loss with different learning rates

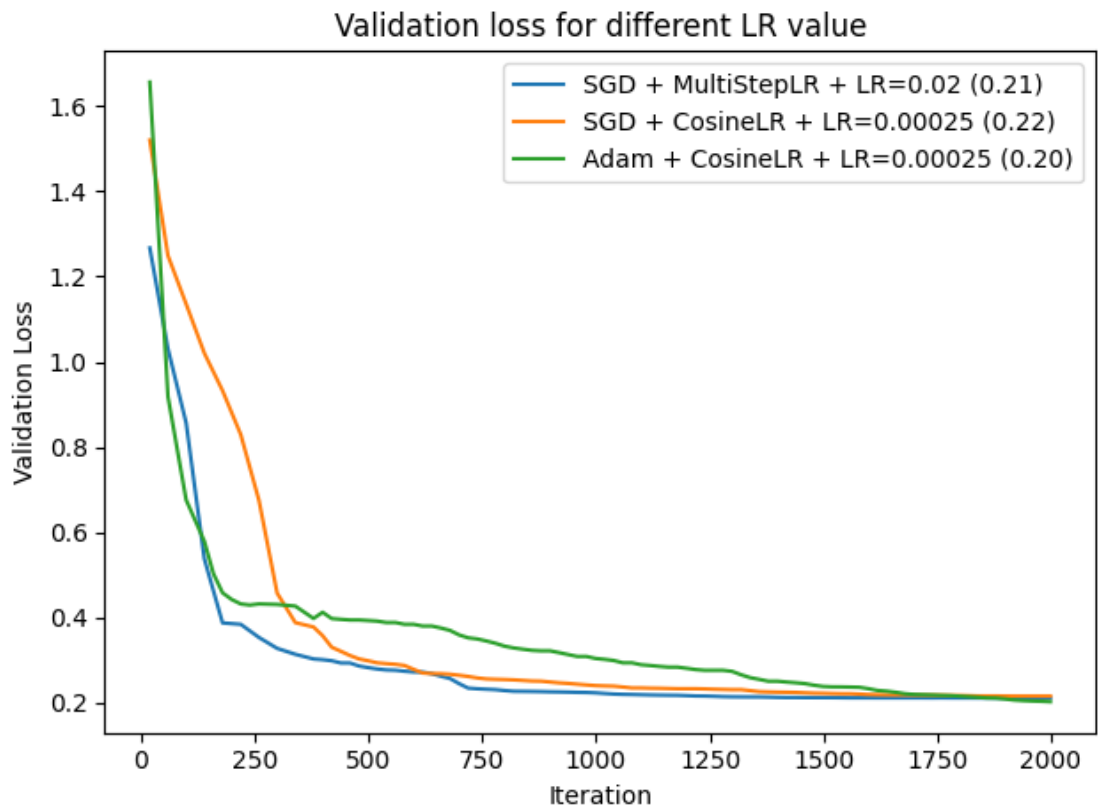


Figure 5.12: Comparison of validation loss with different learning rates

As discussed in section 3.4.4, the model is considered to fit if the training loss and the validation loss are approximately the same value (Chollet, 2017; Goodfellow et al., 2016). The graph displaying both the training loss (0.19) and the validation loss (0.20) is presented in Figure 5.13.

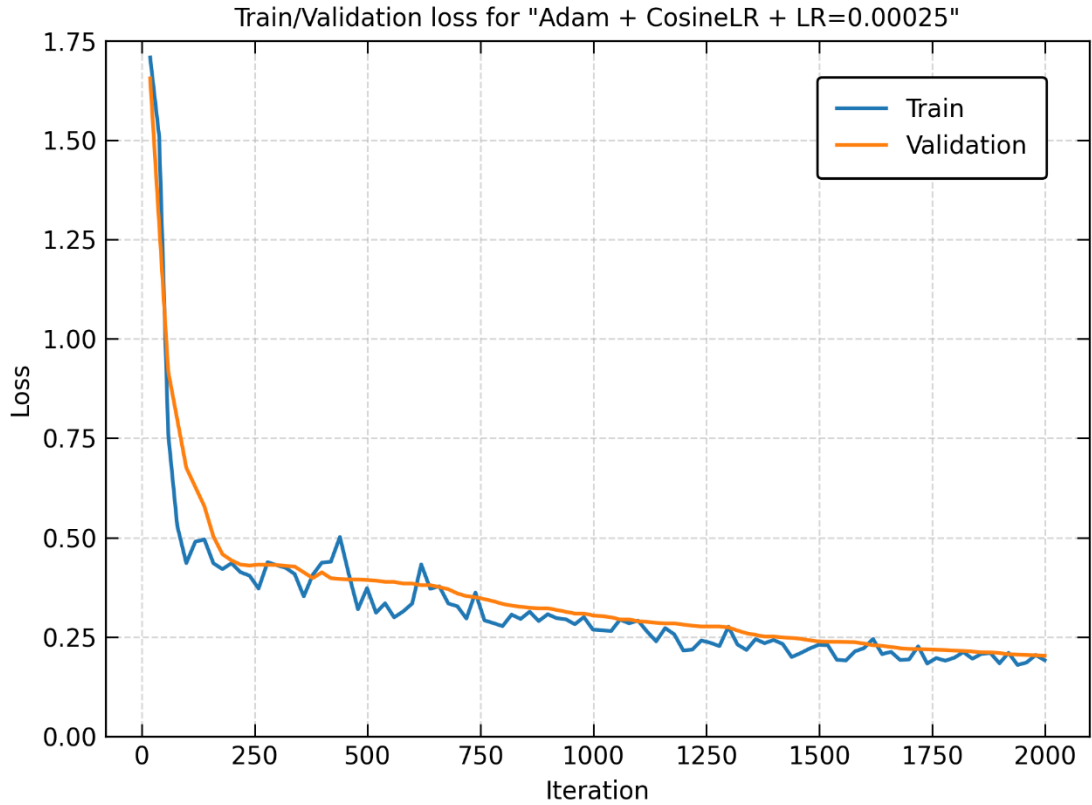


Figure 5.13: Training loss and validation loss of the model

Although the training loss is slightly lower than the validation loss, this model has lower loss values compared to the other two models. Therefore, the model with the **learning rate 0.00025, optimised with Adam optimiser and CosineLR scheduler**, along with the **training hyperparameters** in Table 5.1 was selected as the optimised Mask R-CNN model for the current study. The loss of the optimised Mask R-CNN as defined in the form of the sum of the bounding-box loss (L_{box}), classification loss (L_{cls}), and the mask loss (L_{mask}) (K. He et al., 2017) is shown graphically in Figure 5.14.

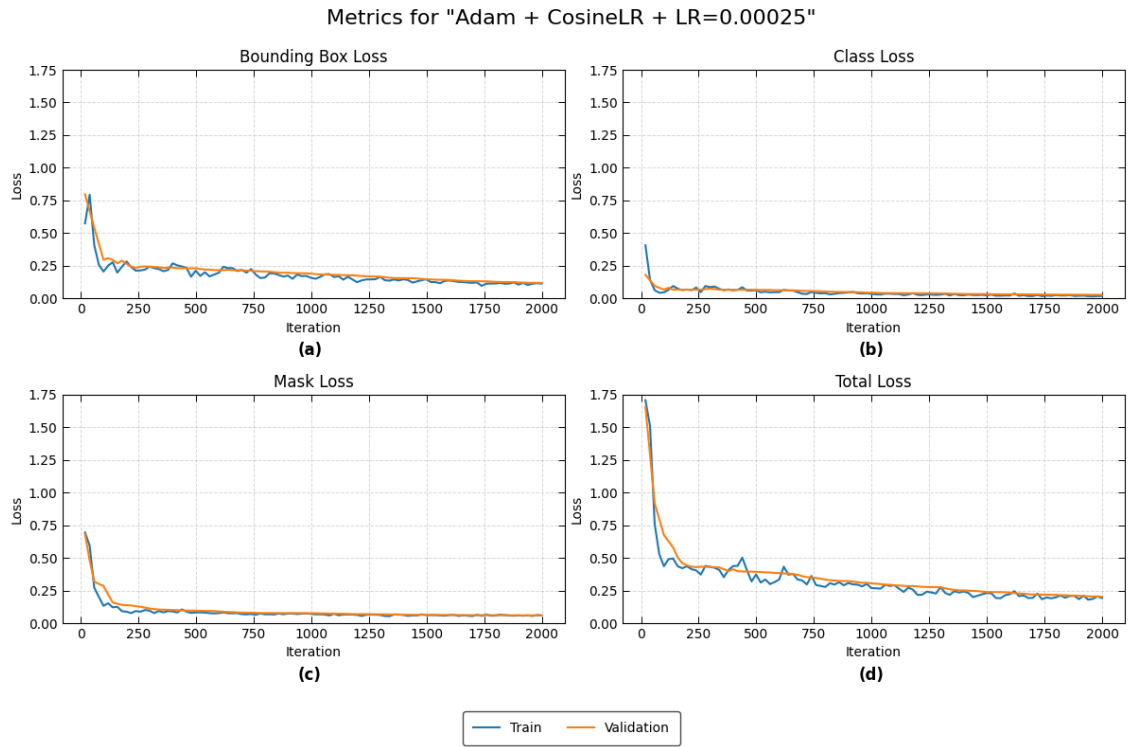


Figure 5.14: Loss: a) Bounding box loss; b) Class loss; c) Mask loss; d) Total loss

When detecting the framing, insulation and drywall installation instances in Figure 5.14(a), the training loss and validation loss seem to slightly deviate from each other until 1,750 iterations. However, from 1,750 to 2,000 iterations, the loss values are approximately the same. It can be further inferred that the as-built classifications of framed, insulated and drywall installed classes have the same lower training and validation loss values closer to 0 at the end of 2,000 iterations in Figure 5.14(b). The mask loss, which is the most crucial value for calculating the pixel mask area, has approximately the same training and validation loss after 500 iterations indicating the robustness of the model in Figure 5.14(c). The total loss values (training loss and validation loss) as shown in Figure 5.13 are the same values as indicated in Figure 5.14(d).

The test images from Project 3 were loaded to the dataset folder. The images were read from the test set one by one as shown by the code in Figure 5.15. The inference speed was 3.2 seconds per image.

```
# Inference using the trained model
cfg.OUTPUT_DIR = "./experiments/"
cfg.MODEL.WEIGHTS = os.path.join(cfg.OUTPUT_DIR, "model_final.pth")
cfg.MODEL.ROI_HEADS.SCORE_THRESH_TEST = 0.5
predictor = DefaultPredictor(cfg)
```

Figure 5.15: Inferring using the trained model

The segmented output images were saved to the outputs folder with the as-built states of framed, insulated and drywall installed automatically recognised. The testing of indoor images using the optimised Mask R-CNN model is demonstrated in Figure 5.16.

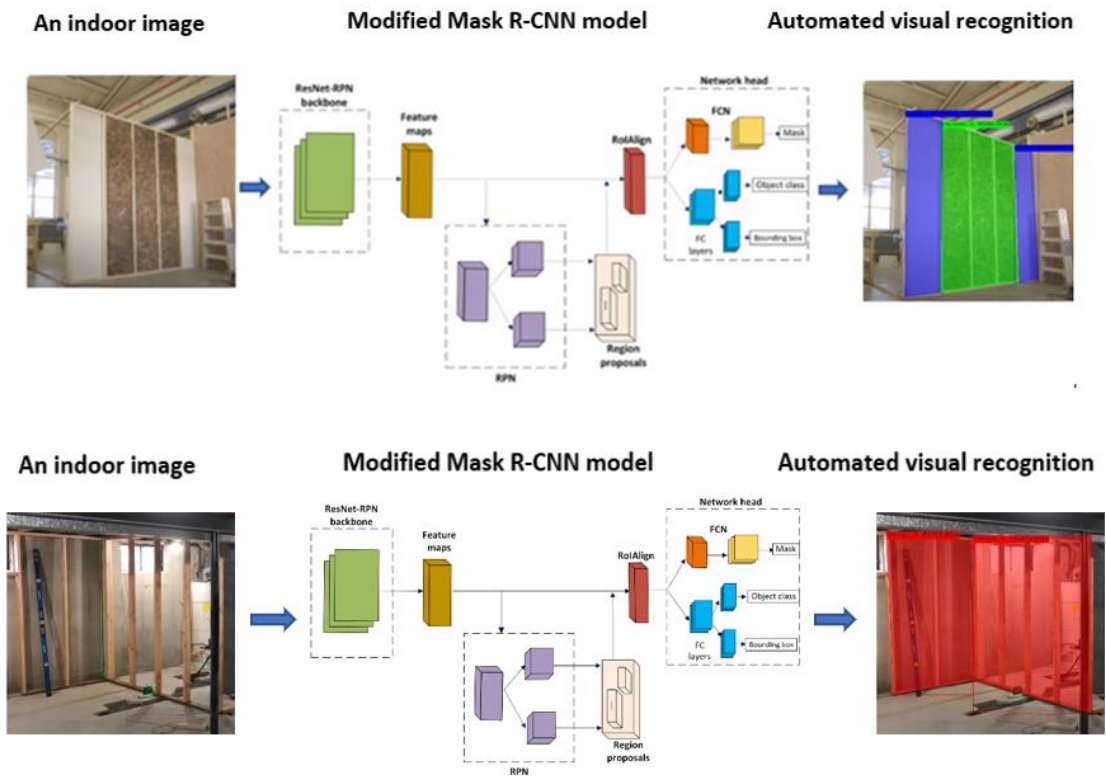


Figure 5.16: Examples of indoor as-built state recognition using Mask R-CNN

Two indoor images were tested using the optimised Mask R-CNN model and Figure 5.16 shows that the as-built states of framing, insulation and drywall installation are automatically detected, and pixel masks are assigned based on the object class. The “framed” object class is assigned with a red coloured pixel mask, while “insulated” has a green coloured pixel mask and “drywall_installed” has a blue coloured pixel mask. These results contributed to achieving **Objective 2** of this research, which is to develop an optimised deep learning model to minimise the impacts of technical challenges for visual recognition of as-built indoor elements. This deep learning model was built using transfer learning and hyperparameter fine-tuning and trained on Google Colab to reduce

computational complexities. The generated pixel masks were used to calculate work-in-progress of the aforementioned as-built states in section 5.2.

5.2 Calculating the work-in-progress of indoor elements

Mask R-CNN is capable of extracting the as-built area of the walls in pixel level masks in the indoor site images. The optimised Mask R-CNN model was post-processed to calculate the area of the pixel masks corresponding to each as-built state detected. The work-in-progress completion of an indoor wall can be calculated using the ratio between the as-built area of the state detected (framing, insulation, and drywall installation) on a given date and the physical wall area obtained from 3D construction models. Extraction of the area of the pixel masks is essential to map the relationship between the pixel areas of the detected element's as-built state and the as-built area of the wall completed on a given date. The underlying idea is to present the indoor site images with their detected as-built state and their estimated work completion ratio. This relationship is mapped as below in Equation 5.1 and 5.2.

$$\frac{Pd}{Pw} = \frac{Sd}{Sw} \quad \text{Equation 5.1}$$

$$Sd = \frac{Pd}{Pw} * Sw \quad \text{Equation 5.2}$$

Where,

Pd = pixel mask area of the detected as-built state on a given date d_i (in pixels)

Pw = pixel mask area of the entire wall detected on a given date d_i (in pixels)

Sd = as-built area of the wall completed on a given date d_i (in m^2)

Sw = physical wall area obtained from the construction models (in m^2)

It can be discerned from Equation 5.1 that the pixel ratio ($\frac{Pd}{Pw}$) equals the work-in-progress completion ratio ($\frac{Sd}{Sw}$). An example of the pixel masks to understand the area of the ($\frac{Pd}{Pw}$) calculation is presented in Figure 5.17.

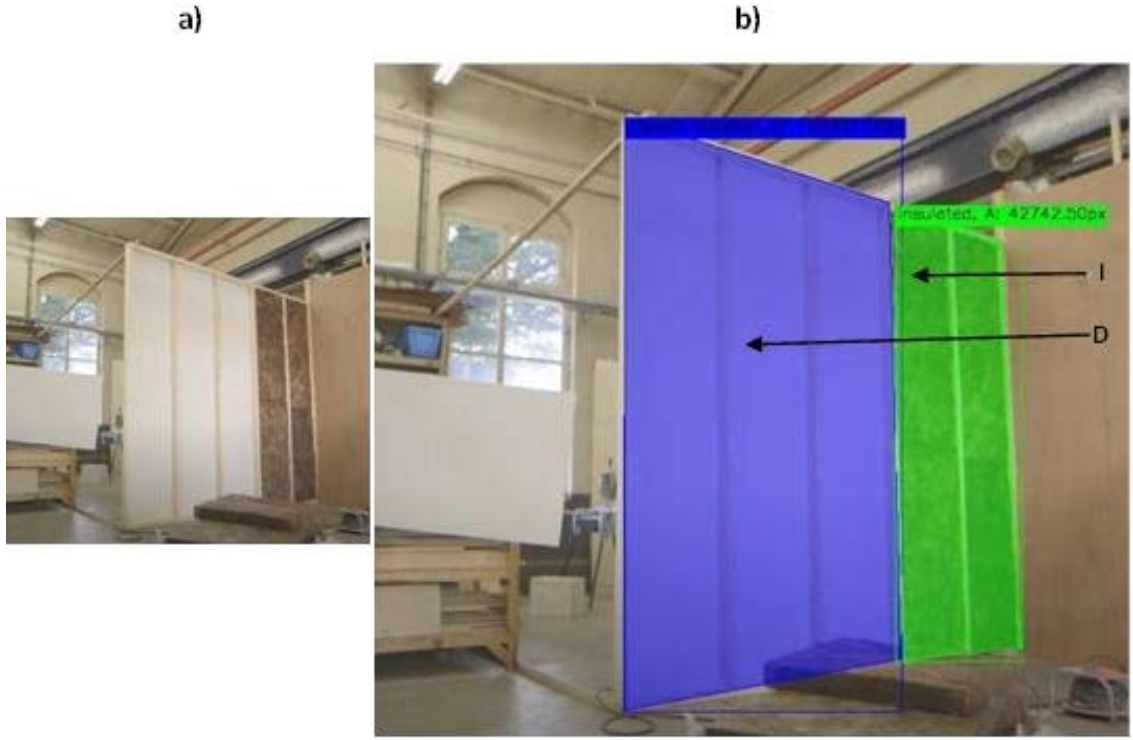


Figure 5.17: a) Image of an indoor wall; b) Segmented mask area calculations

Each segmented pixel mask corresponding to the as-built state has a colour assigned to it, based on the object class label. The “framed” class is a red coloured pixel mask and the “insulated” class is a green coloured pixel mask. The “drywall_installed” class is a blue coloured pixel mask. To calculate the as-built area completion of drywall installed (P_d) in pixels, the area of the segmented mask of the drywall (D) is obtained in pixels. Hence, $P_d = D$. To calculate the pixel mask area of the entire wall, the area of the segmented mask of the drywall (D) and the area of the segmented mask of the insulation (I) are calculated. Therefore, $P_w = D + I$. This relationship can be expressed in Equation 5.3 for further clarification.

$$\frac{\text{Area of the as-built region (drywall)}}{\text{Area of the entire wall}} = \frac{P_d}{P_w} = \frac{D}{D+I} \quad \text{Equation 5.3}$$

As per Equations 5.2 and 5.3, the actual as-built area of the wall (S_d) on a given date can be calculated by multiplying the pixel ratio by the as-planned physical wall area (S_w). This data can be captured from the 3D construction drawings and models obtained from the building contractor. Equation 5.4 can be used for the physical area calculation.

$$\text{Physical area of the as-built region of the wall} = \frac{P_d}{P_w} * S_w(\text{m}^2) \quad \text{Equation 5.4}$$

The process of automatically calculating the area of the segmented pixel mask for each as-built state can be explained by referring to the optimised Mask R-CNN model, which was post-processed. Although the post-processing process was supported by different Python based coding, the code executed on Colab in the notebook “pixel area and ratios calculations” was simplified for the purposes of explanation in the steps discussed below. Numerous OpenCV modules were used for the post-processing process.

Step 1

A folder named “MaskRCNN-workspace” was created in Google Drive, which is linked to the same Google account mentioned in section 5.1. The “MaskRCNN-workspace” folder contains the best trained weight of the optimised Mask R-CNN model, which is referred to as the trained model. A subfolder named “compute_area” was uploaded to the main folder “MaskRCNN-workspace”. The codes related to pixel mask area calculation and ratio calculation were uploaded to “compute_area”. Google Drive was mounted to the Colab notebook named “pixel area and ratios calculations”. Upon mounting Google Drive, the path to the directories of trained_model and compute_area was connected as shown in Figure 5.18.

```
# Mount the Google Drive with the MaskRCNN-workspace sub-folder
from google.colab import drive
drive.mount('/content/drive')
# Path to the main Google Drive folder
# "/content/drive/MyDrive/MaskRCNN-workspace"

# Path to the sub-folder which contains the trained Mask R-CNN model
%cd /content/drive/MyDrive/MaskRCNN-workspace/trained_model

# Path to the sub-folder which contains instructions for area calculation
%cd /content/drive/MyDrive/MaskRCNN-workspace/compute_area
```

Figure 5.18: Mounting Google Drive

Step 2

The libraries and the custom modules needed to execute the code were uploaded. The code is shown in Figure 5.18.

```

# Import the libraries and the custom modules
import os
import cv2
from PIL import Image
from google.colab.patches import cv2_imshow

# custom modules
from compute_area import SegmentationMapDrawer

```

Figure 5.19: Importing the libraries

Step 3

To test the pixel area and ratio calculation ability, site images from Project 3 were uploaded to the “images” subfolder. The output images generated with pixel mask area with ratio calculations were saved in the “outputs” subfolder. This process is shown in Figure 5.20.

```

# Path to the test images
images_dir = "./images"

# Path to the output images with area calculations
output_dir = "./outputs"

```

Figure 5.20: Test images and output directories

Step 4

The “trained_model” was used to generate the masks, bounding box and the class IDs corresponding to each as-built state. As mentioned earlier, these masks of red, green, and blue corresponded to “framed”, “insulated” and “drywall_installed” classes respectively. The bounding box provides the position of the as-built state detected, while the object ID is the identification of the as-built category. The mask provides the pixel segmentation of the as-built state. The segmented boundary of each pixel mask was extracted using the OpenCV module, “Segmentation Map Drawer”. The contours of the segmented mask were used for the pixel area calculation. The OpenCV module of “Contour Area” was used for the area calculation. Accordingly, it was established that the segmented mask area equals the contour area in pixels. The OpenCV module of “put text” was used to display the pixel area in text format in the image corresponding to each mask. The ratio calculations were done through “compute_ratio” using the interpretation in Equation 5.3. The code is presented in Figure 5.21.

```

# Extracting the segmented area and calculating the ratios of pixel mask areas
def draw_segmented_map_cv(image, class_ids, boxes, masks, class_names, compute_ratio, colors):
    if compute_ratio:
        ratio_dict = {'framed': 0.0, 'insulated': 0.0, 'drywall_installed': 0.0}

    height, width = image.shape[:2]
    for i, (class_id, box) in enumerate(zip(class_ids, boxes)):
        contours = utils.get_mask_contours(masks[:, :, i])[0]

        # 2. Calculate area
        mask_area = cv2.contourArea(contours)
        image = draw_mask(image, [contours], colors[class_id])
        if compute_ratio:
            ratio_dict[class_names[class_id]] += mask_area
        pad = 10
        y1, x1, y2, x2 = box
        text = "{}, A: {:.2f}px".format(class_names[class_id], mask_area)

        cv2.rectangle(image, (x1, y1), (x2, y2), colors[class_id], 1)
        (tw, th), _ = cv2.getTextSize(text, cv2.FONT_HERSHEY_SIMPLEX, 0.5, 1)
        cv2.rectangle(image, (x1, y1), (x1 + tw + 2, y1 + th + pad - 3), colors[class_id], -1)

        cv2.polylines(image, [contours], True, colors[class_id], 1)

        cv2.putText(image, text, (x1 + 2, y1 + 12),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 0), 1, cv2.LINE_4)
    if compute_ratio:
        if ratio_dict["drywall_installed"] == 0.0:
            # ratio_dict.pop("drywall_installed")
            ratio = ratio_dict["insulated"] / (
                ratio_dict["insulated"] + ratio_dict["framed"])
            text = "I:I+F = {:.4f}".format(ratio)
        if ratio_dict["insulated"] != 0.0 and ratio_dict["drywall_installed"] != 0.0:
            # ratio_dict.pop("insulated")
            ratio = ratio_dict["drywall_installed"] / (
                ratio_dict["drywall_installed"] + ratio_dict["insulated"])
            text = "D:D+I = {:.4f}".format(ratio)

        if ratio_dict["insulated"] == 0.0:
            # ratio_dict.pop("insulated")
            ratio = ratio_dict["drywall_installed"] / (
                ratio_dict["drywall_installed"] + ratio_dict["framed"])
            text = "D:D+F = {:.4f}".format(ratio)

        (tw, th), _ = cv2.getTextSize(text, cv2.FONT_HERSHEY_SIMPLEX, 0.8, 2)
        cv2.rectangle(image, (width - 150, height - 100), (width - 100 + tw + 2, height - 100 + th + pad - 3), (255, 255, 255), -1)
        cv2.putText(image, text, (width - 150 + 2, height - 100 + 12),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 0), 1, cv2.LINE_4)

    return image

```

Figure 5.21: Pixel masks area calculation

The wall completion percentage as a ratio of the area of drywall installed to the area of the entire wall is **73.87%**. It is shown in Figure 5.22(a). After automatically calculating the work completion percentage, it is matched with the wall area captured from 3D construction drawings/models (AutoCAD), which were provided by the building contractor of Project 3. An extract of the indoor wall is shown in Figure 5.22(b). The physical as-built area of the wall on a given date is calculated as in Equation 5.5.

$$As_{built\ wall\ area} = Work\ completion\ percentage * As_{planned\ wall\ area} \quad \text{Equation 5.5}$$

The as-planned area presented in square metres (m²) in the 3D model as shown in Figure 5.21b is 9.5 m². Therefore, the as-built area of the wall on **26 April 2022** as calculated from Equation 5.5 is **7.02 m²**.

(a)



(b)

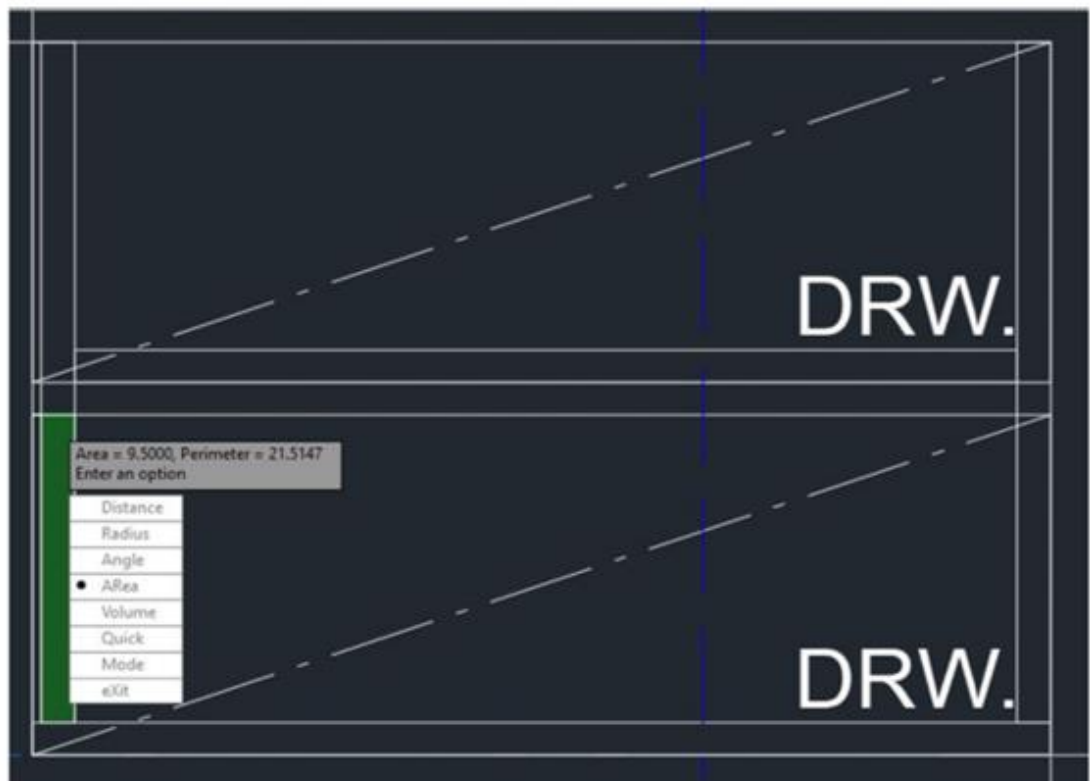


Figure 5.22: a) Work completion; b) As-planned wall area from CAD model

The wall completion is concluded with the drywall sheets being fully installed on **27 April 2022**. This is shown in Figure 5.23.



Figure 5.23: Indoor partition wall is fully installed

In the current study, the optimised Mask R-CNN model was initially employed for automated visual recognition of indoor as-built elements. The same model was post-processed using OpenCV based libraries on Colab to automatically calculate the work-in-progress percentages of the indoor as-built elements. The results contributed to achieving **Objective 3** of this PhD study, which is to automatically calculate the work-in-progress of indoor elements by using an optimised deep learning model.

5.3 Discussion

This chapter examined how the Mask R-CNN model can be optimised by transfer learning and hyperparameter fine-tuning. After training on Colab, this optimised deep learning model was used for automated visual recognition of indoor as-built elements. The current study only focused on the progress monitoring scenario of framing, insulation, and installation of drywalls of indoor walls. The current model with the optimisation improvements through learning rate (0.00025) and CosineLR scheduler with Adam optimiser and the training efficiency improvements through the batch size, network

resolution and number of iterations, reported a mAP of 88.02%, and training and validation losses of 0.19 and 0.2 respectively on a dataset of 2,000 images. The model was trained on Colab to reduce computational complexities and enhance the shareability and replicability of the codes and their results. The optimised Mask R-CNN model was post-processed to automatically calculate the work-in-progress of indoor elements from the same progress monitoring scenario. The model is capable of extracting the areas of the segmented pixel masks corresponding to the as-built states detected and the entire indoor wall in the region of interest. This information was computed to calculate the work-in-progress percentages of indoor elements.

The previous studies of Golparvar-Fard et al. (2015) and Zhang et al. (2009) explained that the traditional practices of as-built state recognition and work-in-progress calculation of construction elements using eye measurement, tape and red pencil lack accuracy, are inefficient and costly compared to computer vision based methods. In Chapter 4, it was revealed that compared to exterior sites, the computer vision based as-built visual recognition from indoor site images is hindered by the technical challenges related to indoor objects, lighting conditions and camera positioning. Using image processing machine learning algorithms, automated visual analysis has been performed for indoor as-built construction elements in some previous studies (Deng et al., 2020; Hamledari & McCabe, 2016; Hamledari, McCabe, & Davari, 2017; Kropp et al., 2012; Kropp et al., 2014; Kropp et al., 2018). These existing studies have exploited distinctive features related to characteristics such as shape, colour, and texture of indoor objects (Hamledari & McCabe, 2016). In some studies, such as Deng et al. (2020), Hamledari, McCabe, Davari and Shahi (2017) and Kropp et al. (2018), the as-built progress details were integrated with the 4D BIM model to visualise progress deviations. However, these methods have high levels of manual intervention. On the other hand, although BIM can help monitor progress, some construction stages are usually not modelled in BIM (Pal et al., 2022), particularly the indoor construction activities (W. Wei et al., 2022; Ying & Lee, 2019). As evidenced from the case projects of the current study, for small and medium scale renovation, refurbishment and fit-out projects, BIMs were not available. Compared to previous studies, the current study established that the application of computer vision technology with deep learning algorithms could be beneficial for automated visual recognition and work-in-progress calculation of as-built indoor elements, mainly in small and medium scale projects when a prior model such as BIM is

not available. Through the optimised and post-processed Mask R-CNN model of the current study, calculation, and visualisation of the as-built completion percentage of an indoor element have been made possible only by presenting the model with an as-built image of the indoor element on a given date.

The recent developments in computer vision based indoor construction progress monitoring have harnessed the detection and classification ability of Mask R-CNN for recognising indoor elements with irregular shapes (Shamsollahi et al., 2021; Ying & Lee, 2019) and the segmentation ability of Mask R-CNN has been used to quantify work completion (W. Wei et al., 2022). Compared to the current study, the manual intervention in the data collection using a handheld Zed2 camera is higher in the study of W. Wei et al. (2022). Nevertheless, it should be acknowledged that the position tracking ability of the Zed2 camera enables the 2D images to be streamlined with 3D BIM through an algorithmic process proposed by them. Unlike the present study, the algorithmic pre-processing presented by W. Wei et al. (2022), in terms of perspective transformation is a mandatory manual intervention since the as-built wall components are mapped with Revit components. The brightness equalisation pre-processing technique of their study required manual intervention because the brightness exposure of one image is different from another image. Although W. Wei et al. (2022) reported a mAP of 96.8%, for the optimised Mask R-CNN model through transfer learning and fine-tuning hyperparameters in TensorFlow framework, they trained the model on only 500 images of brick laying and plastering, and the model was tested on images captured from the same building. This increases the risk of model overfitting. Comparatively, the present study has used a different progress monitoring scenario, which is the framing, insulation and drywall installation captured from two different case projects and tested on a third case project. The two pre-processing methods presented in the current study, the perspective transformation and low illumination enhancement, are not mandatory steps. The hyperparameters fine-tuned in the current study are different from W. Wei et al. (2022)'s methods of loss function improvement, SGD optimiser improvement, and cosine annealing. The present study uses the Adam optimiser with 0.00025 learning rate and CosineLR scheduler. The training workflow of the method is presented through Colab virtual machine on a PyTorch framework unlike the W. Wei et al. (2022) method of using a physical computer with a Tensorflow framework.

Although the competitive advantages of applying deep learning models for indoor construction progress monitoring are highlighted in the studies of Shamsollahi et al. (2021), W. Wei et al. (2022), and Ying and Lee (2019), these studies also point out the complexities in training models. As a solution, the current study offers a training workflow on cloud enabled virtual machine runtime using Google Colab to reduce the computational resource requirements and to enable sharing among project stakeholders. In addition, this study also provides guidance on post-processing the optimised Mask R-CNN on Colab using OpenCV modules to segment the pixel masks of the detected as-built states and calculate the area of them. The indoor wall completion percentage captured using pixel mask area calculations can be matched with the wall areas captured from 3D drawings and models to determine the actual as-built wall area on a given date. This method is an improvement to the handcrafted approaches implemented by Hamledari, McCabe, Davari and Shahi (2017) and Kropp et al. (2018) for work completion calculations. Upon providing training images of other indoor as-built scenes, the automated visual recognition ability and the segmented pixel area calculation ability of the optimised Mask R-CNN model can be extended to other indoor as-built scenes.

5.4 Chapter summary

The results presented in this chapter contribute to achieving **Objective 2** and **Objective 3** of this PhD research. In fulfilling Objective 2, an optimised Mask R-CNN for automated visual recognition of as-built indoor elements was developed using transfer learning and hyperparameter fine-tuning. This method reduced the configuration difficulties associated with developing deep learning models from scratch. Furthermore, this model was trained on Google Colab to reduce the high computing resource requirement. Accordingly, an optimised deep learning model by reducing the computational complexities was presented through this research. Upon providing training images, the model can be used for robust automated visual recognition of as-built indoor elements by minimising the impacts of technical challenges. The optimised Mask R-CNN model was post-processed using OpenCV modules to calculate the areas of the segmented pixel masks corresponding to as-built states recognised. These segmented pixel masks enabled automatically calculating the work-in-progress of indoor elements. Thereby, Objective 3 was achieved. This chapter also presented the artefact encompassing the steps of the design science research framework. These steps are determining the requirements,

developing the artefact, and testing the artefact. In achieving these objectives, the artefact produced in accordance with the design science research framework is digital in nature.

6 Conclusions and recommendations

This chapter concludes the thesis by summarising research objectives achievement, outcomes, and the theoretical and practical contributions of this study. After discussing the limitations of the current study, future research directions are recommended. Section 6.1 revisits how the research objectives are accomplished. Section 6.2 discusses the theoretical and practical contributions of this research. After highlighting the limitations of the study, section 6.3 offers recommendations on future research directions.

6.1 Achieving the research objectives and summary of the outcomes

The gaps in the existing computer vision based indoor construction progress monitoring studies have been identified in this study. Deficiencies in the current understanding of the technical challenges in indoor sites and their impacts impede robust visual recognition of indoor as-built elements. The existing solutions of automated visual recognition for indoor elements mostly depend on handcrafted machine learning algorithms. Although the detection and classification ability of deep learning models can be harnessed to devise robust solutions by minimising the impacts of technical challenges, the computational complexities associated with deep learning are challenging. In addition to automated visual recognition, facilitating the automated calculation of work-in-progress of indoor as-built elements through the deep learning model contributes to improving automation in computer vision based indoor construction progress monitoring. These gaps in the existing studies were reviewed and the central research problem of the present study was formulated as how to improve automation in computer vision based indoor construction progress monitoring. Thus, the aim of this study was to develop a deep learning based approach for improving automation in computer vision based indoor construction progress monitoring.

This thesis achieves three research objectives: (1) investigating the technical challenges and their impacts for visual recognition of indoor as-built elements, (2) developing an optimised deep learning model for automated visual recognition of as-built indoor elements by reducing the computational complexities, and (3) automatically calculating the work-in-progress of indoor elements by post-processing the optimised deep learning model.

Based on the nature of the research problem and the aim, which focuses on addressing a practical problem by developing a technology based artefact, the design-oriented framework of design science research was selected as the underpinning methodological framework. The design science research framework embodies understanding a practical problem by designing and developing an artefact to address the problem, testing the artefact, and communicating the results. This framework was mapped with the research methods to achieve the objectives of this research. In the subsequent sections, outcomes are presented corresponding to each objective.

6.1.1 Investigating the technical challenges and their impacts for indoor as-built elements visual recognition

Achieving this objective contributes to identifying the factors impeding automation in computer vision based indoor construction progress monitoring. Through the systematic review of the previous studies on computer vision based indoor construction progress monitoring in sections 2.4 and 2.5.1, it was established that compared to exterior environments, automatic visual recognition of as-built elements is more challenging in indoor sites. As per the first step of the design science research methodological framework, understanding the practical problem in the real-life context was achieved using indoor site images from the case projects as empirical evidence. The impacts of the technical challenges for visual recognition of as-built indoor elements were demonstrated using these indoor site images.

Technical challenges related to indoor objects, lighting conditions and camera positioning

The challenges identified through the systematic literature review were categorised into technical challenges related to indoor objects, lighting conditions and camera positioning. To corroborate and modify the literature findings on the technical challenges, the empirical evidence gathered during the process of image data collection from the three indoor case projects was used in section 4.1. As evidenced from the empirical findings, the inherent characteristics of the indoor objects and the poor illumination in the indoor sites constrain automatic feature extraction. Handcrafted machine learning algorithms rely on human intervention for feature extraction and thus impede automated object recognition. Camera position plays an important role in data capturing. Although fixed cameras with time-lapse photography are effective in capturing data, if they are not

strategically positioned, the region of interest of the image is prone to distortions. Since these challenges constrain robust object detection, classification and segmentation, the automated visual recognition of as-built state and percentage work completion calculations using progress images are greatly affected. This in turn hinders the automation of monitoring of indoor construction progress.

Significance of time-lapse photography for progress data collection

Since the impacts of occlusions in the background and clutter in indoor sites are different from site to site, applying pre-processing to remove clutter and occlusions related background noise requires heavy manual intervention. However, this study proved that time-lapse photography generated many overlapping images of the progress monitoring scenario to reduce the impacts of background noise caused by clutter and occlusions in indoor sites.

Pre-processing using perspective transformation and low illumination enhancement

Although it is not mandatory, the application of algorithmic pre-processing techniques can enhance the visual quality of the poor images that are impacted by low illumination and camera positioning in indoor construction sites. Therefore, a perspective transformation solution for correcting the distortions of the region of interest affected by camera positioning and an illumination enhancement solution for the images captured under low illuminated scenes were presented using Python programming codes in section 4.2. While deep learning algorithms are proven to be accurate and efficient in terms of feature extraction, a certain level of manual intervention in pre-processing may be required depending on the visual quality of the indoor site images.

In summary, when achieving Objective 1, Chapter 4 explored the nature of the three types of technical challenges and their impacts on robust feature extraction from the indoor images. Two pre-processing algorithmic solutions were tested and presented for when the visual quality of the indoor images is poor.

6.1.2 Developing an optimised deep learning algorithm for automated visual recognition of indoor as-built elements

Objective 2 of this research contributes to developing an optimised Mask R-CNN model for automated visual recognition of indoor as-built elements. Compared to handcrafted machine learning approaches, the deep learning models are more robust in automated object recognition from the indoor site images by reducing the impacts of challenges

related to indoor objects, lighting conditions and camera positioning with minimum human interventions. This is mainly because deep learning models train on thousands of images to extract features and learn patterns compared to the machine learning algorithms, in which the feature extraction is done by the programmer by selecting handcrafted features of texture, edges, colours and shapes.

Modifications to the hyperparameters of the base Mask R-CNN model

This study presented a Mask R-CNN model built using transfer learning and hyperparameter fine-tuning on a ResNet50-FPN backbone. Through experimentation, it was found that the original learning rate of 0.02 needed to be changed to 0.0025 with the optimiser changed from SGD to Adam to improve the accuracy in terms of mAP and loss. The learning rate scheduler was changed from a MultiStepLR to CosineLR to improve the mAP and reduce the loss. The training efficiency of the model was improved by modifying the batch size, network resolution and number of iterations for the three object classes in the dataset. The transfer learning and hyperparameter fine-tuning was found to be an efficient strategy as it requires less adjustment of the configurations compared to building the deep learning model from scratch, which requires defining and assigning layers, weights, activation functions and optimisations.

The improvements of the optimised Mask R-CNN model

The mAP of the optimised Mask R-CNN model is 88.02% with a training loss of 0.19 and a validation loss of 0.20. This model was used to recognise the as-built states of framing, insulation, and drywall installation. The model was tested on an indoor construction project to demonstrate the performance. The model was trained with a very efficient execution time of 3.5 hours on Colab. The inference speed is 3.2 seconds, if a test image is given as the input.

Significance of Colab based training workflow

The model was trained on Colab, and the workflow proved that the use of Colab reduces the training complexities and high computing resource requirements. The use of Colab enhances the shareability and ease of replicability of the deep learning model. The process was extensively explained in section 5.1 of Chapter 5.

To summarise, by achieving Objective 2, this study offers a simplified, efficient, and shareable optimised Mask R-CNN model. The training workflow, which uses the Colab engine, can be replicated in other indoor as-built scenes upon providing training images.

6.1.3 Automatically calculating the percentage work completion of indoor elements by post-processing the optimised deep learning algorithm

To achieve this objective, the optimised Mask R-CNN model was post-processed using OpenCV image processing libraries. The as-built area of the indoor walls in pixel level masks was extracted from the images. This process was explained in section 5.2 of Chapter 5.

Mapping the relationship between the pixel area and the physical area of the wall

By achieving this objective, when an indoor site image is provided, the detected as-built state and the estimated work completion percentage on a given date can be automatically calculated. The instance segmentation ability of Mask R-CNN enables the extraction of the pixel masks of the detected element's as-built state (i.e., framing, insulation, and drywall installation). Each pixel mask has a distinguished colour assigned to them as red, green, and blue for the object class IDs "framed", "insulated" and "drywall_installed" respectively. Using OpenCV image processing libraries, the area of the segmented pixel masks was calculated in pixels. Extraction of the area of the segmented pixel masks is essential to map the relationship between the pixel areas of the detected element's as-built state and the as-built physical area of the wall completed on a given date.

Replicating in similar studies upon providing training images of the scenario

The current study demonstrated how the ratio of the as-built pixel area of the mask "drywall_installed" to the pixel area of the entire wall, which is the sum of pixel areas of the masks "insulated" and "drywall_installed" was calculated. This value is the work completion ratio of an indoor wall on a given date. The same relationship can be replicated in similar studies upon providing the training images of the as-built scenario.

Calculating the actual percentage wall completion through a test case

This study further explained and demonstrated through a test case that the actual as-built area of the wall on a given date was calculated by multiplying the work completion ratio using the physical wall area captured from the 3D construction drawings and models provided by the building contractor.

In summary, achieving Objective 2 and Objective 3 as explained in Chapter 5 followed the steps of designing and developing an artefact to address the problem, testing the artefact, and communicating the results of the design science research framework.

6.2 Theoretical contributions and practical implications

The contributions of this research to the body of knowledge and the practical implications for the construction industry are discussed in the following subsections.

6.2.1 Contributions to the body of knowledge

The contributions of this thesis for research in the construction automation, construction management and built environment domains are threefold.

First, while computer vision applications with digital cameras and deep learning algorithms are well developed in the domains of computer science and software engineering, through this PhD study, the tremendous potential of computer vision is harnessed to improve progress monitoring in construction projects. The current findings are focused on identification of the key research gaps in the existing automated construction progress monitoring literature. With a history of slightly over a decade, the applications of computer vision for indoor construction environment are scarce. This study offers an extensive review of computer vision based indoor construction progress monitoring studies from the perspectives of advances in indoor images collection and visualisation, and automated activity state recognition of indoor works. The evolution of research related to computer vision based indoor construction progress monitoring as discussed in this thesis pinpoints the gaps in the current body of knowledge to better inform future research in the area.

Second, this research contributes to advancing the understanding of the challenges impeding automated visual recognition of indoor as-built elements. When developing algorithmic solutions to extract information from indoor site images, the understanding of the technical challenges and their impacts of indoor as-built visual recognition is advantageous for construction automation researchers. The indoor objects and lighting related challenges are not always within the control of the researcher to rectify. This research informs the importance of selecting the strategic vantage point for installing cameras if fixed cameras are used for image collection. The constraints and consequences related to camera positioning are investigated specially when the data collection takes place during a global pandemic or a similarly challenging situation with zero or minimum access to construction sites. The pre-processing algorithmic solutions presented in this thesis can be replicated in similar studies to enhance the visual quality of the images, in

situations where the images are affected by low illumination and camera positioning related distortions.

As the third contribution, this study offers a simplified, efficient, and shareable Mask R-CNN model with a training workflow on a virtual machine platform such as Google Colab. This optimised deep learning model, and the training workflow can be replicated in similar studies for object detection, classification, and segmentation related construction problems upon providing the training images of the studied scenario. Some potential scenarios could be construction equipment monitoring, worker monitoring and activity recognition. Beyond construction problems, the model has a whole range of potential uses in built environment applications. One such example is building defects analysis such as crack detection and measurement of the dimensions of the cracks using the optimised Mask R-CNN model and the training workflow on Colab.

6.2.2 Implications for the construction industry

The present study makes several noteworthy contributions to the construction industry, which can be explained in three aspects.

First, in construction projects, the traditional methods of calculating work completion percentage involving a tape measure and red pencil are laborious, costly and error prone. This study paves the way to automate construction work-in-progress calculation solutions by providing a Mask R-CNN based model as an alternative approach. With the Mask R-CNN model presented in this study, the construction professionals such as contractors, site supervisors, site engineers and project managers can determine the percentage work completion on a given day, when a construction image captured from an on-site camera is processed. Since the current study focused on the indoor construction environment, the small and medium contractors who are usually involved with refurbishment, renovation and fit-out projects can primarily benefit from this type of automated, cost effective and accurate progress information extraction even without a prior model such as BIM. In practice, detailed BIMs are not widely generated for small and medium scale fit-out, renovation and refurbishment projects compared to mega and large scale projects. If the current deep learning model is used for work completion measurement of different as-built scenarios, the annotated training images need to be provided. Although the pixel level annotation can be a tedious task, it can be easily taught to users with a non-

programming background. No software installation is needed as a web based, free-of-charge application such as “make-sense” can be used.

Second, this study offers an accessible and shareable Mask R-CNN on Google Colab for project stakeholders as a potential avenue for automated recognition of construction elements. With only a Google account and without having to spend thousands of dollars on high performance computing resources for on-site progress information extraction, project stakeholders can access and share this model through cloud platforms and Google Drive. Upon providing training images, the accurate detection, classification, and segmentation ability of the optimised Mask R-CNN model can be extended to recognise as-built scenes other than the indoor walls. One such application could be automated visual recognition and work completion percentage calculation of prefabricated walls construction. When this model is implemented on construction progress monitoring applications, the users only need a Google account to access Colab and Google Drive. Colab and Google Drive are both free-of-charge cloud based applications. Therefore, no software installations are required. However, a reliable internet connection is necessary. Depending on the user requirements, Colab offers versions spanning from US\$10 per month. The storage capacity of Google Drive can be extended to 100 GB for about US\$2 per month. Since the optimised Mask R-CNN model is readily executable and shareable, there is no additional programming training required to run it on Colab and to store the images on Google Drive. The study provides an easy-to-follow technical flow diagram and further explanation on the workflow of the deep learning model to be understood by construction professionals with a non-programming background.

Third, this study provides a better understanding for construction professionals on using time-lapse cameras as a data collection method. As a method for collecting progress images, this has substantially lower costs, takes less time, and requires less configuration of the equipment. The Brinno TLC200 Pro camera used in this study can be purchased for about US\$220. No training is required because such cameras have easy-to-follow configuration settings. No special software is needed to process the images collected.

The contributions of this PhD study as discussed above are summarised and shown in Figure 6.1 in terms of the contribution to knowledge (in the upper part) and the implications for the construction industry (in the lower part).

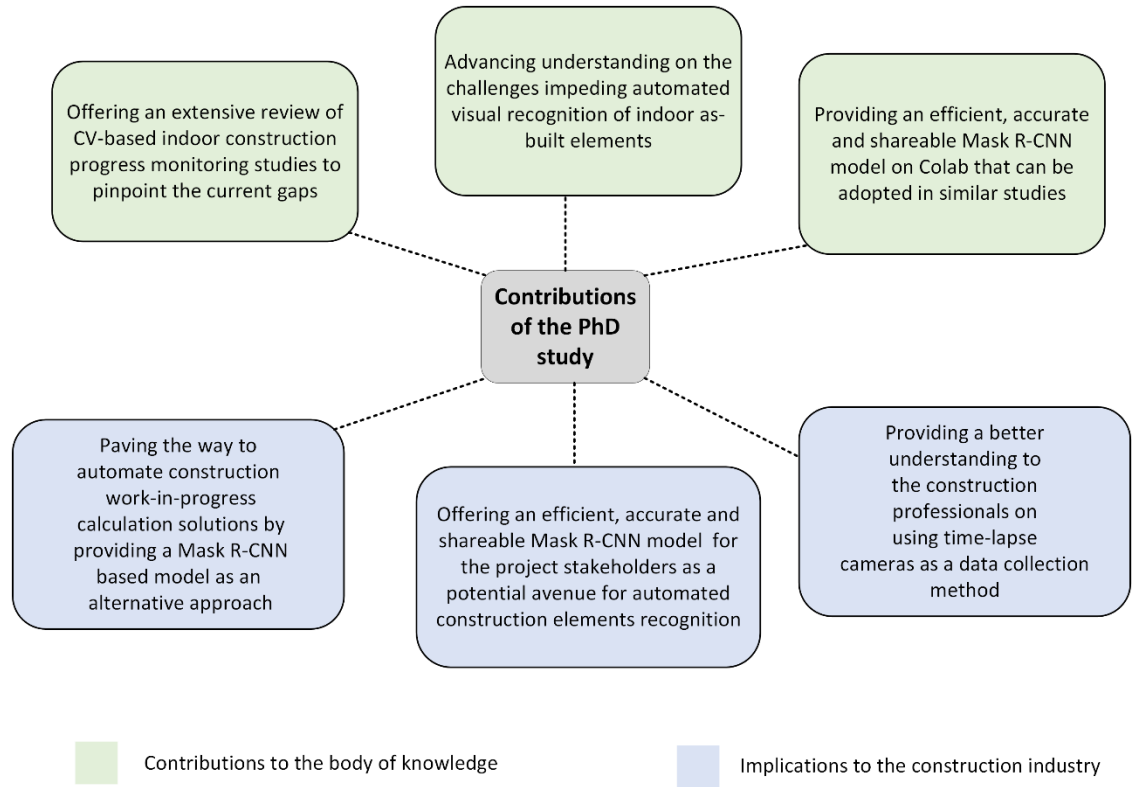


Figure 6.1: Contributions of the PhD study

6.3 Limitations of the current study and future research directions

Several limitations of the current study need to be acknowledged to recommend future research directions.

The case projects and the progress monitoring scenario selected in the current study contributed to investigating the technical challenges and developing an optimised Mask R-CNN model for indoor construction progress monitoring. Due to COVID related restrictions, selecting the number of case studies and the progress monitoring scenarios had some constraints as explained in section 1.5. Therefore, it is recommended that further research should be undertaken using the same training workflow on Colab, and the optimised Mask R-CNN model by widening the case projects and the progress monitoring scenarios. Other segmentation deep learning algorithms can also be trained and tested if different progress monitoring scenarios are explored because it provides more cases for comparison. A comparison of the performance of other segmentation deep learning algorithms with the optimised Mask R-CNN model for each progress monitoring scenario is recommended as a future research direction.

Due to the specific restrictions of the case projects, such as cluttered indoor environments and interruptions to construction workers, and the COVID related restrictions, the most appropriate and practical data collection technology adopted in this study is time-lapse photography on fixed cameras. More research is needed to better understand and demonstrate the integration of image data from cameras with position data captured from technologies such as UWB and RFID and volumetric data from laser scanners for indoor construction progress monitoring. In future studies, data fusion strategies can be used to extract and process information from multiple sensors with different mounting mechanisms such as UAV and UGV.

The current study has successfully demonstrated the application and the performance of the Mask R-CNN model, and it can be replicated in similar problem contexts such as construction safety and productivity monitoring. To be extended to other problem contexts, the training images need to be annotated in the COCO format. Labelling the images with pixel level annotation is a time consuming and tedious task. To resolve the problem of pixel level annotations associated with Mask R-CNN model, future studies should consider approaches such as deep active learning, which is unsupervised learning to automatically evaluate the uncertainty of un-labelled training data, select the most meaningful-to-learn, and thereby train the deep learning model with the least human intervention.

The detailed BIMs were not available in the small and medium scale fit-out, renovation and refurbishment projects selected for this study. In future investigations, if the BIM for the project is available, the application of 5D BIM linked with the cost information could be considered to establish automated progress payments systems.

In addition to the above recommendations, it is noteworthy to point out that there is scope to customise other hyperparameters in the Mask R-CNN model apart from the fine-tuning conducted in the current study. An additional layer of validation can be provided for the optimised Mask R-CNN model through statistical analysis conducted on multiple datasets of different indoor construction progress monitoring scenarios in future studies. If multiple datasets of different progress monitoring scenarios are available, a comparative study between the conventional machine learning models such as Canny edge detector and a further optimised Mask R-CNN model can be conducted across datasets using mAP

as the evaluation metric. By doing so, the automation potential of deep learning algorithms can be further explored in future studies.

In summary, the outcomes of this research have multiple contributions to the body of knowledge and the construction industry. These contributions can further be harnessed in future studies with the recommended research directions.

References

- Abdelsayed, M., & Navon, R. (1999). An information sharing, Internet-based, system for project control. *Civil Engineering and Environmental Systems*, 16(3), 211-233. <https://doi.org/10.1080/02630259908970263>
- Agarwal, R., Chandrasekaran, S., & Sridhar, M. (2016, June 24). *Imagining construction's digital future*. McKinsey & Company. <https://www.mckinsey.com/capabilities/operations/our-insights/imagining-constructions-digital-future>
- Ahankoob, A., Manley, K., Hon, C., & Drogemuller, R. (2021). The influence of building information modelling on the absorptive capacity of project-based organisations. *Architectural Engineering and Design Management*, 1-21. <https://doi.org/10.1080/17452007.2021.1881879>
- Ahmadian Fard Fini, A., Maghrebi, M., Forsythe, P. J., & Waller, T. S. (2022). Using existing site surveillance cameras to automatically measure the installation speed in prefabricated timber construction. *Engineering, Construction and Architectural Management*, 29(2), 573-600. <https://doi.org/10.1108/ECAM-04-2020-0281>
- Akbari, A., Awais, M., Bashar, M., & Kittler, J. (2021). How does loss function affect generalization performance of deep learning? application to human age estimation. In N. Lawrence (Ed.), *International Conference on Machine Learning* (pp. 141-151). PMLR.
- Akinosho, T. D., Oyedele, L. O., Bilal, M., Ajayi, A. O., Delgado, M. D., Akinade, O. O., & Ahmed, A. A. (2020). Deep learning in the construction industry: A review of present status and future innovations. *Journal of Building Engineering*, 32, 101827. <https://doi.org/10.1016/j.jobbe.2020.101827>
- Alizadehsalehi, S., & Yitmen, I. (2018). A concept for automated construction progress monitoring: Technologies adoption for benchmarking project performance control. *Arabian Journal for Science and Engineering*, 44(5), 4993-5008. <https://doi.org/10.1007/s13369-018-3669-1>
- Alter, S. (2008). Defining information systems as work systems: implications for the IS field. *European Journal of Information Systems*, 17(5), 448-469. <https://doi.org/10.1057/ejis.2008.37>
- Amer, F., & Golparvar-Fard, M. (2018a). Decentralized visual 3D mapping of scattered work locations for high frequency tracking of indoor construction activities. In C. Wang, C. Berryman, R. Harris, C. Harper, & Y. Lee (Eds.), *Construction Research Congress 2018: Construction Information Technology* (pp. 491-500). American Society of Civil Engineers. <https://doi.org/10.1061/9780784481264.048>
- Amer, F., & Golparvar-Fard, M. (2018b). Crowd sourced visual data collection for monitoring indoor construction in 3D. In Mohamed. Abdul-Malak, H. Khoury, A. Singh, & S. Yazdani (Eds.), *2nd European and Mediterranean Structural Engineering and Construction Conference*. ISEC Press. <https://doi.org/10.14455/ISEC.res.2018.15>
- Amer, F., Zhao, Z., Tang, S., & Torres, W. (2018). *Constructing Locally Dense Point Clouds Using OpenSfM and ORB-SLAM2*. arXiv. <https://doi.org/10.48550/arXiv.1804.08243>
- Arzarello, F., & Olivero, F. (2006). Theories and empirical researches: towards a common framework. *WORKING GROUP 11 Different theoretical perspectives and approaches in research in mathematics education*.

- Asadi, K., & Han, K. (2018). Real-time image-to-BIM registration using perspective alignment for automated construction monitoring. In C. Wang, C. Berryman, R. Harris, C. Harper, & Y. Lee (Eds.), *Construction Research Congress 2018: Construction Information Technology* (pp. 388-397). American Society of Civil Engineers. <https://doi.org/10.1061/9780784481264.038>
- Asadi, K., & Han, K. (2020). An integrated aerial and ground vehicle (UAV-UGV) system for automated data collection for indoor construction sites. In P. Tang, D. Grau, & M. El Asmar (Eds.), *Construction Research Congress 2020: Computer Applications* (pp. 846-855). American Society of Civil Engineers. <https://doi.org/10.1061/9780784482865.090>
- Asadi, K., Ramshankar, H., Noghabaei, M., & Han, K. (2019). Real-time image localization and registration with bim using perspective alignment for indoor monitoring of construction. *Journal of Computing in Civil Engineering*, 33(5). [https://doi.org/10.1061/\(asce\)cp.1943-5487.0000847](https://doi.org/10.1061/(asce)cp.1943-5487.0000847)
- Avison, D., Baskerville, R., & Myers, M. (2001). Controlling action research projects. *Information technology & people*, 14(1), 28–45. <https://doi.org/10.1108/09593840110384762>
- Bansal, V. K., & Pal, M. (2009). Construction schedule review in GIS with a navigable 3D animation of project activities. *International Journal of Project Management*, 27(5), 532-542. <https://doi.org/10.1016/j.ijproman.2008.07.004>
- Baskerville, R. (2008). What design science is not. *European Journal of Information Systems*, 17(5), 441-443. <https://doi.org/10.1057/ejis.2008.45>
- Baskerville, R., & Myers, M. D. (2004). Special issue on action research in information systems: Making IS research relevant to practice: Foreword. *MIS Quarterly*, 329-335.
- Baskerville, R. (1999). Investigating information systems with action research. *Communications of the Association for Information Systems*, 2(1), 19.
- Behnam, A., Wickramasinghe, D. C., Ghaffar, M. A. A., Vu, T. T., Tang, Y. H., & Isa, H. B. M. (2016). Automated progress monitoring system for linear infrastructure projects using satellite remote sensing. *Automation in Construction*, 68, 114-127. <https://doi.org/10.1016/j.autcon.2016.05.002>
- Benbasat, I., & Zmud, R. W. (2003). The identity crisis within the IS discipline: Defining and communicating the discipline's core properties. *MIS Quarterly*, 183-194.
- Bharati, P., & Pramanik, A. (2020). Deep learning techniques—R-CNN to Mask R-CNN: A survey. In A. K. Das, J. Nayak, B. Naik, S. K. Pati, & D. Pelusi (Eds.), *Computational Intelligence in Pattern Recognition* (pp. 657-668). Springer. <https://doi.org/10.1007/978-981-13-9042-5-56>
- Bhargavi, K., & Jyothi, S. (2014). A survey on threshold based segmentation technique in image processing. *International Journal of Innovative Research and Development*, 3(12), 234-239.
- Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). *Yolov4: Optimal speed and accuracy of object detection*. arXiv. <https://doi.org/10.48550/arXiv.2004.10934>
- Bodenbenner, P., Feuerriegel, S., & Neumann, D. (2013). Design science in practice: designing an electricity demand response system. In J. Brocke, R. Hekkala, S. Ram, & M. Rossi (Eds.), *Design Science at the Intersection of Physical and Virtual Design. DESRIST 2013* (pp. 293-307). Springer. https://doi.org/10.1007/978-3-642-38827-9_20
- Bohn, J. S., & Teizer, J. (2010). Benefits and barriers of construction project monitoring using high-resolution automated cameras. *Journal of Construction Engineering*

- and Management, 136(6), 632-640. [https://doi.org/10.1061/\(ASCE\)CO.1943-7862.0000164](https://doi.org/10.1061/(ASCE)CO.1943-7862.0000164)
- Bolya, D., Zhou, C., Xiao, F., & Lee, Y. J. (2019, October 29 – November 1). *YOLACT: Real-time instance segmentation* [Paper presentation]. IEEE/CVF International Conference on Computer Vision, Seoul, South Korea. <https://doi.org/10.1109/ICCV.2019.00925>
- Bosché, F. (2010). Automated recognition of 3D CAD model objects in laser scans and calculation of as-built dimensions for dimensional compliance control in construction. *Advanced Engineering Informatics*, 24(1), 107-118. <https://doi.org/10.1016/j.aei.2009.08.006>
- Bosché, F. (2012). Plane based registration of construction laser scans with 3D/4D building models. *Advanced Engineering Informatics*, 26(1), 90-102. <https://doi.org/10.1016/j.aei.2011.08.009>
- Bosché, F., Ahmed, M., Turkan, Y., Haas, C. T., & Haas, R. (2015). The value of integrating Scan-to-BIM and Scan-vs-BIM techniques for construction monitoring using laser scanning and BIM: The case of cylindrical MEP components. *Automation in Construction*, 49, 201-213. <https://doi.org/10.1016/j.autcon.2014.05.014>
- Bosché, F., Guillemet, A., Turkan, Y., Haas, C. T., & Haas, R. (2014). Tracking the built status of MEP works: Assessing the value of a Scan-vs-BIM system. *Journal of Computing in Civil Engineering*, 28(4), 05014004. [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000343](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000343)
- Bosche, F., & Haas, C. T. (2008). Automated retrieval of 3D CAD model objects in construction range images. *Automation in Construction*, 17(4), 499-512. <https://doi.org/10.1016/j.autcon.2007.09.001>
- Bosche, F., Haas, C. T., & Akinci, B. (2009). Automated recognition of 3D CAD objects in site laser scans for project 3D status visualization and performance control. *Journal of Computing in Civil Engineering*, 23(6), 311-318. [https://doi.org/doi:10.1061/\(ASCE\)0887-3801\(2009\)23:6\(311\)](https://doi.org/doi:10.1061/(ASCE)0887-3801(2009)23:6(311))
- Braun, A., & Borrmann, A. (2019). Combining inverse photogrammetry and BIM for automated labeling of construction site images for machine learning. *Automation in Construction*, 106, 102879. <https://doi.org/10.1016/j.autcon.2019.102879>
- Braun, A., Tuttas, S., Borrmann, A., & Stilla, U. (2020). Improving progress monitoring by fusing point clouds, semantic data and computer vision. *Automation in Construction*, 116, 103210. <https://doi.org/10.1016/j.autcon.2020.103210>
- Brinno. (2022, March). *Brinno TLC200 Pro HDR Time Lapse Camera*. <https://www.brinno.com/time-lapse-camera/TLC200Pro>
- Canesche, M., Bragança, L., Neto, O. P. V., Nacif, J. A., & Ferreira, R. (2021, May 22-28). *Google Colab CAD4U: Hands-on Cloud Laboratories for Digital Design* [Paper presentation]. 2021 IEEE International Symposium on Circuits and Systems (ISCAS), Daegu, South Korea. <https://doi.org/10.1109/ISCAS51556.2021>
- Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6), 679-698. <https://doi.org/10.1109/TPAMI.1986.4767851>
- Carneiro, T., Da Nóbrega, R. V. M., Nepomuceno, T., Bian, G.-B., De Albuquerque, V. H. C., & Reboucas Filho, P. P. (2018). Performance analysis of Google Colaboratory as a tool for accelerating deep learning applications. *IEEE Access*, 6, 61677-61685. <https://doi.org/10.1109/ACCESS.2018.2874767>

- Civil Aviation Safety Authority. (2022, February). *Drone rules*.
<https://www.casa.gov.au/knownyourdrone>
- Casado-García, Á., Domínguez, C., García-Domínguez, M., Heras, J., Inés, A., Mata, E., & Pascual, V. (2019). CLoDSA: a tool for augmentation in classification, localization, detection, semantic segmentation and instance segmentation tasks. *BMC bioinformatics*, 20(1), 1-14. <https://doi.org/10.1186/s12859-019-2931-1>
- Chaurasia, A., & Culurciello, E. (2017, December 10-17). *Linknet: Exploiting encoder representations for efficient semantic segmentation* [Paper presentation]. 2017 IEEE Visual Communications and Image Processing (VCIP), St. Petersburg, Florida, United States. <https://doi.org/10.1109/VCIP.2017.8305148>
- Chen, M.-Y., Chiang, H.-S., Lughofer, E., & Egrioglu, E. (2020). Deep learning: emerging trends, applications and research challenges. *Soft Computing*, 24(11), 7835-7838. <https://doi.org/10.1007/s00500-020-04939-z>
- Cheng, T., Venugopal, M., Teizer, J., & Vela, P. A. (2011). Performance evaluation of ultra wideband technology for construction resource location tracking in harsh environments. *Automation in Construction*, 20(8), 1173-1184. <https://doi.org/10.1016/j.autcon.2011.05.001>
- Chi, S., & Caldas, C. H. (2011). Automated object identification using optical video cameras on construction sites. *Computer-aided Civil and Infrastructure Engineering*, 26(5), 368-380. <https://doi.org/10.1111/j.1467-8667.2010.00690.x>
- Cho, S. I., & Kang, K. (2010). A Low-Complexity 128-Point Mixed-Radix FFT Processor for MB-OFDM UWB Systems. *ETRI Journal*, 32(1), 1-10. <https://doi.org/10.4218/etrij.10.0109.0232>
- Chollet, F. (2017). *Deep learning with Python*. Simon and Schuster.
- Dai, F., Rashidi, A., Brilakis, I., & Vela, P. (2012). Comparison of image based and time-of-flight-based technologies for 3D reconstruction of infrastructure. *Journal of Construction Engineering and Management*, 139(1), [https://doi.org/10.1061/\(ASCE\)CO.1943-7862.0000565](https://doi.org/10.1061/(ASCE)CO.1943-7862.0000565)
- Deng, H., Hong, H., Luo, D., Deng, Y., & Su, C. (2020). Automatic indoor construction process monitoring for tiles based on BIM and computer vision. *Journal of Construction Engineering and Management*, 146(1). [https://doi.org/10.1061/\(asce\)co.1943-7862.0001744](https://doi.org/10.1061/(asce)co.1943-7862.0001744)
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009, June 20-25). *ImageNet: A large-scale hierarchical image database* [Paper presentation]. 2009 IEEE conference on computer vision and pattern recognition (CVPR), Miami, Florida, United States. <https://doi.org/10.1109/CVPR.2009.5206848>
- Ekanayake, B. (2022). A deep learning based building defects detection tool for sustainability monitoring. In Y.G. Sandanayake, S. Gunatilake, & K.G.A.S. Waidyasekara (Eds.), *10th World Construction Symposium* (pp 2-13). Colombo, Sri Lanka. <https://doi.org/10.31705/WCS.2022.1>
- El-Omari, S., & Moselhi, O. (2011). Integrating automated data acquisition technologies for progress reporting of construction projects. *Automation in Construction*, 20(6), 699-705. <https://doi.org/10.1016/j.autcon.2010.12.001>
- Ergen, E., & Akinci, B. (2007, September 5-6). *An overview of approaches for utilizing RFID in construction industry* [Paper presentation]. 2007 1st Annual RFID Eurasia, Istanbul, Turkey. <https://doi.org/10.1109/RFIDEURASIA.2007.4368087>
- Ergen, E., Akinci, B., & Sacks, R. (2007). Tracking and locating components in a precast storage yard utilizing radio frequency identification technology and

- GPS. *Automation in Construction*, 16(3), 354-367.
<https://doi.org/10.1016/j.autcon.2006.07.004>
- Fang, W., Ding, L., Zhong, B., Love, P. E. D., & Luo, H. (2018). Automated detection of workers and heavy equipment on construction sites: A convolutional neural network approach. *Advanced Engineering Informatics*, 37, 139-149.
<https://doi.org/10.1016/j.aei.2018.05.003>
- Fard, M. G., & Peña-Mora, F. (2007). Application of visualization techniques for construction progress monitoring. In L. Soibelman & B. Akinci (Eds.), *Computing in Civil Engineering 2007* (pp. 216-223). American Society of Civil Engineers. [https://doi.org/10.1061/40937\(261\)27](https://doi.org/10.1061/40937(261)27)
- Fellows, R. F., & Liu, A. M. (2015). *Research methods for construction* (4th ed.). John Wiley & Sons.
- Fetić, A., Jurić, D., & Osmanković, D. (2012, May 21-25). *The procedure of a camera calibration using camera calibration toolbox for MATLAB* [Paper presentation]. 35th International Convention MIPRO, Opatija, Croatia.
- Feurer, M., Hutter, F. (2019). Hyperparameter Optimization. In F. Hutter, L. Kotthoff, & J. Vanschoren (Eds.), *Automated Machine Learning* (pp. 3-33). Springer.
https://doi.org/10.1007/978-3-030-05318-5_1
- Golparvar-Fard, M., Bohn, J., Teizer, J., Savarese, S., & Peña-Mora, F. (2011). Evaluation of image-based modeling and laser scanning accuracy for emerging automated performance monitoring techniques. *Automation in Construction*, 20(8), 1143-1155. <https://doi.org/10.1016/j.autcon.2011.04.016>
- Golparvar-Fard, M., Peña-Mora, F., & Savarese, S. (2011). Integrated sequential as-built and as-planned representation with D4AR tools in support of decision-making tasks in the AEC/FM Industry. *Journal of Construction Engineering and Management*, 137(12), 1099-1116. [https://doi.org/10.1061/\(ASCE\)CO.1943-7862.0000371](https://doi.org/10.1061/(ASCE)CO.1943-7862.0000371)
- Golparvar-Fard, M., Peña-Mora, F., & Savarese, S. (2015). Automated progress monitoring using unordered daily construction photographs and IFC based Building Information Models. *Journal of Computing in Civil Engineering*, 29(1). [https://doi.org/10.1061/\(asce\)cp.1943-5487.0000205](https://doi.org/10.1061/(asce)cp.1943-5487.0000205)
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*.
- Google Research. (2022, May). *Colaboratory*.
<https://research.google.com/colaboratory/faq.html>
- Gregor, S., & Hevner, A. R. (2013). Positioning and presenting design science research for maximum impact. *MIS quarterly*, 337-355.
- Gu, J., Wang, Z., Kuen, J., Ma, L., Shahrudiy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J., & Chen, T. (2018). Recent advances in convolutional neural networks. *Pattern Recognition*, 77, 354-377.
<https://doi.org/10.1016/j.patcog.2017.10.013>
- Gulli, A., & Pal, S. (2017). *Deep learning with Keras*. Packt Publishing Ltd.
- Guo, F., Qian, Y., Rizos, D., Suo, Z., & Chen, X. (2021). Automatic rail surface defects inspection based on Mask R-CNN. *Transportation research record*, 2675(11), 655-668. <https://doi.org/10.1177/03611981211019034>
- Guo, X., Li, Y., & Ling, H. (2016). LIME: Low-light image enhancement via illumination map estimation. *IEEE Transactions on Image Processing*, 26(2), 982-993. <https://doi.org/10.1109/TIP.2016.2639450>
- Gurevich, U., & Sacks, R. (2014). Examination of the effects of a KanBIM production control system on subcontractors' task selections in interior works. *Automation in Construction*, 37, 81-87. <https://doi.org/10.1016/j.autcon.2013.10.003>

- Ham, Y., Han, K., Lin, J. J., & Golparvar-Fard, M. (2016). Visual monitoring of civil infrastructure systems via camera-equipped Unmanned Aerial Vehicles (UAVs): a review of related works. *Visualization in Engineering*, 4(1). <https://doi.org/10.1186/s40327-015-0029-z>
- Hamledari, H., & McCabe, B. (2016). Automated visual recognition of indoor project-related objects: Challenges and solutions. In J.L. Perdomo-Rivera, A. González-Quevedo, C. López del Puerto, F. Maldonado-Fortunet, & O. I. Molina-Bas (Eds.), *Construction Research Congress 2016: Old and New Construction Technologies Converge in Historic San Juan* (pp. 2573-2582). American Society of Civil Engineers. <https://doi.org/10.1061/9780784479827.256>
- Hamledari, H., McCabe, B., & Davari, S. (2017). Automated computer vision based detection of components of under-construction indoor partitions. *Automation in Construction*, 74, 78-94. <https://doi.org/10.1016/j.autcon.2016.11.009>
- Hamledari, H., McCabe, B., Davari, S., & Shahi, A. (2017). Automated schedule and progress updating of IFC based 4D BIMs. *Journal of Computing in Civil Engineering*, 31(4). [https://doi.org/10.1061/\(asce\)cp.1943-5487.0000660](https://doi.org/10.1061/(asce)cp.1943-5487.0000660)
- Hamledari, H., McCabe, B., Davari, S., Shahi, A., Rezazadeh Azar, E., & Flager, F. (2017, May 31-June 3). *Evaluation of computer vision and 4D BIM based construction progress tracking on a UAV platform* [Paper presentation]. 6th CSCE/ASCE/CRC International Construction Specialty Conference, Vancouver, Canada.
- Hamledari, H., Rezazadeh Azar, E., & McCabe, B. (2018). IFC-Based Development of As-Built and As-Is BIMs Using Construction and Facility Inspection Data: Site-to-BIM Data Transfer Automation. *Journal of Computing in Civil Engineering*, 32(2). [https://doi.org/10.1061/\(asce\)cp.1943-5487.0000727](https://doi.org/10.1061/(asce)cp.1943-5487.0000727)
- Han, J., Zhang, D., Cheng, G., Liu, N., & Xu, D. (2018). Advanced deep learning techniques for salient and category-specific object detection: a survey. *IEEE Signal Processing Magazine*, 35(1), 84-100. <https://doi.org/10.1109/MSP.2017.2749125>
- Han, K., & Golparvar-Fard, M. (2017). Crowdsourcing BIM-guided collection of construction material library from site photologs. *Visualization in Engineering*, 5(1). <https://doi.org/10.1186/s40327-017-0052-3>
- Han, K., Cline, D., & Golparvar-Fard, M. (2015). Formalized knowledge of construction sequencing for visual monitoring of work-in-progress via incomplete point clouds and low-LoD 4D BIMs. *Advanced Engineering Informatics*, 29(4), 889-901. <https://doi.org/10.1016/j.aei.2015.10.006>
- He, D., Qiu, Y., Miao, J., Zou, Z., Li, K., Ren, C., & Shen, G. (2022). Improved Mask R-CNN for obstacle detection of rail transit. *Measurement*, 190, 110728. <https://doi.org/10.1016/j.measurement.2022.110728>
- He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017, October 22-29). *Mask R-CNN* [Paper presentation]. 2017 IEEE International Conference on computer vision, Venice, Italy. <https://doi.org/10.1109/ICCV.2017.322>
- Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 75-105.
- Hong, Y., Hammad, A. W. A., Sepasgozar, S., & Akbarnezhad, A. (2019). BIM adoption model for small and medium construction organisations in Australia. *Engineering, Construction and Architectural Management*, 26(2), 154-183. <https://doi.org/10.1108/ECAM-04-2017-0064>
- Howard, R., Restrepo, L., & Chang, C.-Y. (2017). Addressing individual perceptions: An application of the unified theory of acceptance and use of technology to

- building information modelling. *International Journal of Project Management*, 35(2), 107-120. <https://doi.org/10.1016/j.ijproman.2016.10.012>
- Hua, J., Hao, T., Zeng, L., & Yu, G. (2021). YOLOMask, an instance segmentation algorithm based on complementary fusion network. *Mathematics*, 9(15), 1766. <https://doi.org/10.3390/math9151766>
- Hubbard, B., Wang, H., Leasure, M., Ropp, T., Lofton, T., Hubbard, S., & Lin, S. (2015, April 22-25). *Feasibility study of UAV use for RFID material tracking on construction sites* [Paper presentation]. 51st ASC Annual International Conference, Texas, United States.
- Iannizzotto, G., & Vita, L. (2000). Fast and accurate edge based segmentation with no contour smoothing in 2D real images. *IEEE Transactions on Image Processing*, 9(7), 1232-1237. <https://doi.org/10.1109/83.847835>
- Ibrahim, A., Sabet, A., & Golparvar-Fard, M. (2019). BIM-driven mission planning and navigation for automatic indoor construction progress detection using robotic ground platform. In J. O'Donnell, A. Chassiakos, D. Rovas, & D. Hall (Eds.), *2019 European Conference on Computing in Construction* (pp. 182-189). European Council on Computing in Construction. <https://doi.org/10.35490/EC3.2019.195>
- Iivari, J., & Venable, J. R. (2009). *Action research and design science research- Seemingly similar but decisively dissimilar* [Paper presentation]. 17th European Conference on Information Systems (ECIS), Verona, Italy.
- imgaug. (2022, June). *Overview of Augmenters*. <https://imgaug.readthedocs.io/en/latest/index.html#>
- Jabbar, H., & Khan, R. Z. (2015). Methods to avoid over-fitting and under-fitting in supervised machine learning (comparative study). In J. Stephen, H. Rohil, & S. Vasavi (Eds.), *Computer Science, Communication and Instrumentation Devices*, 163-172. Research Publishing Services. https://doi.org/10.3850/978-981-09-5247-1_017
- Järvinen, P. (2007). Action research is similar to design science. *Quality & Quantity*, 41(1), 37-54. <https://doi.org/10.1007/s11135-005-5427-1>
- Jian, L., Wang, C., Liu, Y., Liang, S., Yi, W., & Shi, Y. (2013). Parallel data mining techniques on graphics processing unit with compute unified device architecture (CUDA). *The Journal of Supercomputing*, 64(3), 942-967. <https://doi.org/10.1007/s11227-011-0672-7>
- Johannesson, P., & Perjons, E. (2014). *An introduction to design science*. Springer.
- Jönsson, S., & Lukka, K. (2006). There and back again: doing interventionist research in management accounting. *Handbooks of management accounting research*, 1, 373-397. [https://doi.org/10.1016/S1751-3243\(06\)01015-7](https://doi.org/10.1016/S1751-3243(06)01015-7)
- Jorda, M., Valero-Lara, P., & Pena, A. J. (2019). Performance evaluation of cuDNN convolution algorithms on NVIDIA Volta GPUs. *IEEE Access*, 7, 70461-70473. <https://doi.org/10.1109/ACCESS.2019.2918851>
- Kaganami, H. G., & Beiji, Z. (2009). Region based segmentation versus edge detection. In J. Pan, Y. Chen, & L.C. Jain (Eds.), *2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing* (pp. 1217-1221). IEEE Computer Society. <https://doi.org/10.1109/IIH-MSP.2009.13>
- Kang, S.-H., & Tesar, D. (2004, November 13-19). *Indoor GPS metrology system with 3D probe for precision applications* [Paper presentation]. 2004 ASME International Mechanical Engineering Congress and RD&D Expo, Anaheim, California United States.

- Kardovskyi, Y., & Moon, S. (2021). Artificial intelligence quality inspection of steel bars installation by integrating Mask R-CNN and stereo vision. *Automation in Construction*, 130, 103850. <https://doi.org/10.1016/j.autcon.2021.103850>
- Kartika, I., & Mohamed, S. S. (2011). Frame differencing with post-processing techniques for moving object detection in outdoor environment. In M.N. Taib, R. Adnan, A.M. Samad, N.M.Tahir, Z. Hussain, & M.H.F. Rahiman (Eds.), *2011 IEEE 7th International Colloquium on Signal Processing and its Applications (CSPA 2011)* (pp. 172-176). IEEE CSS Chapter Malaysia. <https://doi.org/10.1109/CSPA18771.2011>
- Kasanen, E., Lukka, K., & Siitonen, A. (1993). The constructive approach in management accounting research. *Journal of management accounting research*, 5(1), 243-264.
- Khoreva, A., Benenson, R., Hosang, J., Hein, M., & Schiele, B. (2017, July 21-26). *Simple does it: Weakly supervised instance and semantic segmentation* [Paper presentation]. IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, Hawaii, United States. <https://doi.org/10.1109/CVPR.2017.181>
- Khoury, H., Chdid, D., Oueis, R., Elhajj, I., & Asmar, D. (2015). Infrastructureless approach for ubiquitous user location tracking in construction environments. *Automation in Construction*, 56, 47–66. <https://doi.org/10.1016/j.autcon.2015.04.009>
- Kim, C., Son, H., & Kim, C. (2013). Automated construction progress measurement using a 4D building information model and 3D data. *Automation in Construction*, 31, 75-82. <https://doi.org/10.1016/j.autcon.2012.11.041>
- Kim, J., Chi, S., & Seo, J. (2018). Interaction analysis for vision based activity identification of earthmoving excavators and dump trucks. *Automation in Construction*, 87, 297-308. <https://doi.org/10.1016/j.autcon.2017.12.016>
- Kim, M.-K., Wang, Q., Park, J.-W., Cheng, J. C. P., Sohn, H., & Chang, C.-C. (2016). Automated dimensional quality assurance of full-scale precast concrete elements using laser scanning and BIM. *Automation in Construction*, 72, 102-114. <https://doi.org/10.1016/j.autcon.2016.08.035>
- Kim, P., Park, J., & Cho, Y. (2019). As-is geometric data collection and 3D visualization through the collaboration between UAV and UGV. In M. Al-Hussein (Ed.), *36th International Symposium on Automation and Robotics in Construction (ISARC 2019)* (pp. 544-551). The International Association for Automation and Robotics in Construction. <https://doi.org/10.22260/ISARC2019/0073>
- Kingma, D. P., & Ba, J. (2014). *Adam: A method for stochastic optimization*. arXiv. <https://doi.org/10.48550/arXiv.1412.6980>
- Ko, C.-H. (2013). 3D-Web-GIS RFID location sensing system for construction objects. *The Scientific World Journal*. 2013(3). <http://doi.org/10.1155/2013/217972>
- Koo, B., & Fischer, M. (2000). Feasibility study of 4D CAD in commercial construction. *Journal of Construction Engineering and Management*, 126(4), 251-260. [https://doi.org/10.1061/\(ASCE\)0733-9364\(2000\)126:4\(251\)](https://doi.org/10.1061/(ASCE)0733-9364(2000)126:4(251))
- Kopsida, M., Brilakis, I., & Vela, P. A. (2015, October 27-29). *A review of automated construction progress monitoring and inspection methods* [Paper presentation]. 32nd CIB W78 Conference 2015, Eindhoven, Netherlands.
- Krawatzek, R., Hofmann, M., Jacobi, F., & Dinter, B. (2013). Constructing software intensive methods: A design science research process with early feedback cycles. In J. Brocke, R. Hekkala, S. Ram, & M. Rossi (Eds.), *Design Science at*

- the Intersection of Physical and Virtual Design. DESRIST 2013* (pp. 486-493). Springer. https://doi.org/10.1007/978-3-642-38827-9_41
- Kropp, C., Koch, C., König, M., & Brilakis, I. (2012, June 27-29). *A framework for automated delay prediction of finishing works using video data and BIM based construction simulation* [Paper presentation]. 14th International Conference on Computing in Civil and Building Engineering, Moscow, Russia.
- Kropp, C., König, M., & Koch, C. (2013). Object recognition in BIM registered videos for indoor progress monitoring. In G. Suter, Y. Rafiq & P. de Wilde (Eds.), *20th EG-ICE International Workshop on Intelligent Computing in Engineering* (pp. 12-21). European Group For Intelligent Computing in Engineering (eg-ice).
- Kropp, C., Koch, C., & König, M. (2014). Drywall state detection in image data for automatic indoor progress monitoring. In R.R Issa, & I. Flood (Eds.), *Computing in Civil and Building Engineering (2014)* (pp. 347-354). American Society of Civil Engineers. <https://doi.org/10.1061/9780784413616.044>
- Kropp, C., Koch, C., & König, M. (2016, July 6-8). *Model based pose estimation for visual indoor progress monitoring using line features* [Paper presentation]. 16th International Conference on Computing in Civil and Building Engineering, Osaka, Japan.
- Kropp, C., Koch, C., & König, M. (2018). Interior construction state recognition with 4D BIM registered image sequences. *Automation in Construction*, 86, 11-32. <https://doi.org/10.1016/j.autcon.2017.10.027>
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-444. <https://doi.org/10.1038/nature14539>
- Lee, K. W., & Park, J. K. (2019). Comparison of characteristics of 3D laser scanner for construction of indoor geospatial information. *International Journal of Advanced Science and Technology*, 129, 151-160. <https://doi.org/10.33832/ijast.2019.129.13>
- Lee, S. H., Peña-Mora, F., & Park, M. (2006). Dynamic planning and control methodology for strategic and operational construction project management. *Automation in Construction*, 15(1), 84-97. <https://doi.org/10.1016/j.autcon.2005.02.008>
- Li, Y., Lu, Y., & Chen, J. (2021). A deep learning approach for real-time rebar counting on the construction site based on YOLOv3 detector. *Automation in Construction*, 124, 103602. <https://doi.org/10.1016/j.autcon.2021.103602>
- Li, Y., Qi, H., Dai, J., Ji, X., & Wei, Y. (2017, July 21-26). *Fully convolutional instance-aware semantic segmentation* [Paper presentation]. 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, Hawaii, United States. <https://doi.org/10.1109/CVPR.2017.472>
- Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017, July 21-26). *Feature pyramid networks for object detection* [Paper presentation]. 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, Hawaii, United States. <https://doi.org/10.1109/CVPR.2017.106>
- Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft COCO: Common Objects in Context. In D. Fleet, T. Pajdla, B. Schiele, & T. Tuytelaars (Eds.), *Computer vision – ECCV 2014* (pp. 740-755). Springer. https://doi.org/10.1007/978-3-319-10602-1_48
- Lin, Y.-C., Cheung, W.-F., & Siao, F.-C. (2014). Developing mobile 2D barcode/RFID-based maintenance management system. *Automation in Construction*, 37, 110-121. <https://doi.org/10.1016/j.autcon.2013.10.004>

- Lindholm, A. L. (2008). A constructive study on creating core business relevant CREM strategy and performance measures. *Facilities*, 26(7), 343–358. <https://doi.org/10.1108/02632770810877976>
- Liu, C., M.E. Sepasgozar, S., Shirowzhan, S., & Mohammadi, G. (2021). Applications of object detection in modular construction based on a comparative evaluation of deep learning algorithms. *Construction Innovation*, 22(1), 141-159. <https://doi.org/10.1108/CI-02-2020-0017>
- Liu, J., Zhang, Q., Li, X., Li, G., Liu, Z., Xie, Y., Li, K., & Liu, B. (2021). Transfer learning-based strategies for fault diagnosis in building energy systems. *Energy and Buildings*, 250, 111256. <https://doi.org/10.1016/j.enbuild.2021.111256>
- Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., & Pietikäinen, M. (2020). Deep learning for generic object detection: A survey. *International Journal of computer vision*, 128(2), 261-318. <https://doi.org/10.1007/s11263-019-01247-4>
- Liu, L. Y., Stumpf, A., Chin, S., Ganeshan, R., & Hicks, D. (1995, July 5-8). *Construction daily log management system using multimedia technology* [Paper presentation]. 2nd Congress on Computing in Civil Engineering, New York, United States.
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). Ssd: Single Shot Multibox Detector. In B. Leibe, J. Matas, N. Sebe, & M. Welling (Eds.), *Computer Vision – ECCV 2016* (pp. 21-37). Springer. https://doi.org/10.1007/978-3-319-46448-0_2
- Liu, W., Sun, J., Li, W., Hu, T., & Wang, P. (2019, Sep 26). Deep Learning on Point Clouds and Its Application: A Survey. *Sensors (Basel)*, 19(19), 4188. <https://doi.org/10.3390/s19194188>
- Love, P. E. D., Tse, R. Y. C., & Edwards, D. J. (2005). Time–cost relationships in Australian building construction projects. *Journal of Construction Engineering and Management*, 131(2), 187-194. [https://doi.org/10.1061/\(asce\)0733-9364\(2005\)131:2\(187\)](https://doi.org/10.1061/(asce)0733-9364(2005)131:2(187))
- Lu, Q., & Lee, S. (2017). Image based technologies for constructing as-is building information models for existing buildings. *Journal of Computing in Civil Engineering*, 31(4). [https://doi.org/10.1061/\(asce\)cp.1943-5487.0000652](https://doi.org/10.1061/(asce)cp.1943-5487.0000652)
- Lukka, K. (2003). The constructive research approach. In L. Ojala & O.-P. Hilmola (Eds.), *Case study research in logistics. Publications of the Turku School of Economics and Business Administration, Series B*, 1(2003) (pp. 83-101). Turku School of Economics and Business Administration.
- Ma, Z., & Liu, S. (2018). A review of 3D reconstruction techniques in civil engineering and their applications. *Advanced Engineering Informatics*, 37, 163-174. <https://doi.org/10.1016/j.aei.2018.05.005>
- Maalek, R., & Sadeghpour, F. (2013). Accuracy assessment of ultra-wide band technology in tracking static resources in indoor construction scenarios. *Automation in Construction*, 30, 170-183. <https://doi.org/10.1016/j.autcon.2012.10.005>
- Martinez, P., Ahmad, R., & Al-Hussein, M. (2019). A vision based system for pre-inspection of steel frame manufacturing. *Automation in Construction*, 97, 151-163. <https://doi.org/10.1016/j.autcon.2018.10.021>
- Martinez, P., Barkokebas, B., Hamzeh, F., Al-Hussein, M., & Ahmad, R. (2021). A vision based approach for automatic progress tracking of floor paneling in offsite construction facilities. *Automation in Construction*, 125, 103620. <https://doi.org/10.1016/j.autcon.2021.103620>

- Master Builders Association New South Wales. (2021, July). *Construction shutdown FAQs*. <https://www.mbansw.asn.au/construction-shutdown-faqs>
- McCabe, B., Hamledari, H., Shahi, A., Zangeneh, P., & Azar, E. R. (2017). Roles, benefits, and challenges of using UAVs for indoor smart construction applications. In K. Lin, N. El-Gohary, & P. Tang (Eds.), *Computing in Civil Engineering 2017: Sensing, Simulation, and Visualization* (pp. 349 – 357). American Society of Civil Engineers.
<https://doi.org/10.1061/9780784480830.043>
- Moragane, H. P. M. N. L. B., Perera, B. A. K. S., Palihakkara, A. D., & Ekanayake, B. (2022). Application of computer vision construction for progress monitoring: a qualitative investigation. *Construction Innovation, ahead-of-print*(ahead-of-print). <https://doi.org/10.1108/CI-05-2022-0130>
- Munawar, H. S., Hammad, A. W., Haddad, A., Soares, C. A. P., & Waller, S. T. (2021). Image based crack detection methods: A review. *Infrastructures*, 6(8), 115.
<https://doi.org/10.3390/infrastructures6080115>
- Nalini, M., & Radhika, K. (2020, October 7-9). *Comparative analysis of deep network models through transfer learning* [Paper presentation]. 4th International Conference on IoT in Social, Mobile, Analytics and Cloud (I-SMAC), Palladam, India. <https://doi.org/10.1109/I-SMAC49090.2020.9243469>
- Nanni, L., Ghidoni, S., & Brahnham, S. (2017). Handcrafted vs. non-handcrafted features for computer vision classification. *Pattern Recognition*, 71, 158-172.
<https://doi.org/10.1016/j.patcog.2017.05.025>
- Navon, R. (2007). Research in automated measurement of project performance indicators. *Automation in Construction*, 16(2), 176-188.
<https://doi.org/10.1016/j.autcon.2006.03.003>
- Navon, R., & Sacks, R. (2007). Assessing research issues in Automated Project Performance Control (APPC). *Automation in Construction*, 16(4), 474-484.
<https://doi.org/10.1016/j.autcon.2006.08.001>
- O'Mahony, N., Campbell, S., Carvalho, A., Harapanahalli, S., Hernandez, G. V., Krpalkova, L., Riordan, D., & Walsh, J. (2019). Deep learning vs. traditional computer vision. In K. Arai, & S. Kapoor (Eds.), *Advances in Computer Vision: Proceedings of 2019 Computer Vision Conference* (pp. 28-144). Springer.
<https://doi.org/10.1007/978-3-030-17795-9>
- Oh, S.-W., Chang, H.-J., Kim, Y.-S., Lee, J.-B., & Kim, H.-S. (2004, September 21-25). *An application of PDA and barcode technology for the improvement of information management in construction projects* [Paper presentation]. 21st International Symposium on Automation and Robotics in Construction (ISARC 2004), Jeju Island, South Korea, <https://doi.org/10.22260/ISARC2004/0090>
- Ohkawara, M., Saito, H., & Fujishiro, I. (2021). Experiencing GPU path tracing in online courses. *Graphics and Visual Computing*, 4, 200022.
<https://doi.org/10.1016/j.gvc.2021.200022>
- Omar, H., Mahdjoubi, L., & Kheder, G. (2018). Towards an automated photogrammetry based approach for monitoring and controlling construction site activities. *Computers in Industry*, 98, 172-182.
<https://doi.org/10.1016/j.compind.2018.03.012>
- Omar, T., & Nehdi, M. L. (2016). Data acquisition technologies for construction progress tracking. *Automation in Construction*, 70, 143-155.
<https://doi.org/10.1016/j.autcon.2016.06.016>

- Oo, B. L., & Lim, B. T. H. (2021). Changes in job situations for women workforce in construction during the COVID-19 pandemic. *Construction Economics and Building*, 21(2). <https://doi.org/10.5130/AJCEB.v21i2.7526>
- OpenCV. (2022a, June). *Canny Edge Detection*. OpenCV. https://docs.opencv.org/4.x/da/d22/tutorial_py_canny.html
- OpenCV. (2022b, June). *Geometric Image Transformations*. OpenCV. https://docs.opencv.org/4.x/da/d6e/tutorial_py_geometric_transformations.html
- Pal, A., & Hsieh, S.-H. (2021). Deep learning based visual data analytics for smart construction management. *Automation in Construction*, 131, 103892. <https://doi.org/10.1016/j.autcon.2021.103892>
- Pal, A., Lin, J. J., & Hsieh, S.-H. (2022). A framework for automated daily construction progress monitoring leveraging unordered site photographs. In R. R. Issa (Ed.), *Computing in Civil Engineering* (pp. 538-545). American Society of Civil Engineers. <https://doi.org/10.1061/9780784483893.067>
- Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10), 1345-1359. <https://doi.org/10.1109/TKDE.2009.191>
- Park, J., Cai, H., & Perissin, D. (2018). Bringing information to the field: Automated photo registration and 4D BIM. *Journal of Computing in Civil Engineering*, 32(2). [https://doi.org/10.1061/\(asce\)cp.1943-5487.0000740](https://doi.org/10.1061/(asce)cp.1943-5487.0000740)
- Peffer, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, 24(3), 45-77. <https://doi.org/10.2753/MIS0742-1222240302>
- Piirainen, K. A., & Gonzalez, R. A. (2013). Seeking constructive synergy: design science and the constructive research approach. In J. Brocke, R. Hekkala, S. Ram, & M. Rossi (Eds.), *Design Science at the Intersection of Physical and Virtual Design. DESRIST 2013* (pp. 59-72). Springer. https://doi.org/10.1007/978-3-642-38827-9_5
- Pinheiro, P. O., Collobert, R., & Dollár, P. (2015). Learning to segment object candidates. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, & R. Garnett (Eds.), *Advances in neural information processing systems 28* (pp. 1990-1998). Association for Computing Machinery. <https://doi.org/10.5555/2969442.2969462>
- Poku, S. E., & Arditi, D. (2006). Construction scheduling and progress control using geographical information systems. 20(5), 351-360. [https://doi.org/10.1061/\(ASCE\)0887-3801\(2006\)20:5\(351\)](https://doi.org/10.1061/(ASCE)0887-3801(2006)20:5(351))
- Pradhananga, N., & Teizer, J. (2013). Automatic spatio-temporal analysis of construction site equipment operations using GPS data. *Automation in Construction*, 29, 107-122. <https://doi.org/10.1016/j.autcon.2012.09.004>
- Praveen Gujjar, J., Prasanna Kumar H, R., & Chiplunkar, N. N. (2021). Image Classification and Prediction using Transfer Learning in Colab Notebook. *Global Transitions Proceedings*, 2(2), 382-385. <https://doi.org/10.1016/j.gltp.2021.08.068>
- Pu, S., & Vosselman, G. (2007). Extracting windows from terrestrial laser scanning. In P. Rönholm, H. Hyyppä, & J. Hyyppä (Eds.), *ISPRS Workshop on Laser Scanning 2007 and SilviLaser 2007* (pp. 320-325). ISPRS.
- Pučko, Z., Šuman, N., & Rebolj, D. (2018). Automated continuous construction progress monitoring using multiple workplace real time 3D scans. *Advanced Engineering Informatics*, 38, 27-40. <https://doi.org/10.1016/j.aei.2018.06.001>

- Rahman, F. H., Newaz, S. H. S., Au, T.-W., Suhaili, W. S., Mahmud, M. A. P., & Lee, G. M. (2022). EnTruVe: ENergy and TRUst-aware Virtual Machine allocation in VEHICLE fog computing for catering applications in 5G. *Future Generation Computer Systems*, 126, 196-210. <https://doi.org/10.1016/j.future.2021.07.036>
- Rawlinsons. (2015). *Australian construction handbook* (33rd ed.). Rawlinson's Publishing.
- Razavi, S., Young, D., Nasir, H., Haas, C., Caldas, C. H., Goodrum, P., & Murray, P. (2008, June 10-13). *Field trial of automated material tracking in construction* [Paper presentation]. Annual Conference of the Canadian Society for Civil Engineering 2008, Quebec, Canada.
- Razavi, S. N., & Haas, C. T. (2010). Multisensor data fusion for on-site materials tracking in construction. *Automation in Construction*, 19(8), 1037-1046. <https://doi.org/10.1016/j.autcon.2010.07.017>
- Razavi, S. N., & Haas, C. T. (2011). Using reference RFID tags for calibrating the estimated locations of construction materials. *Automation in Construction*, 20(6), 677-685. <https://doi.org/10.1016/j.autcon.2010.12.009>
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016, June 27-30). *You Only Look Once: Unified, real-time object detection* [Paper presentation]. 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, Nevada, United States. <https://doi.org/10.1109/CVPR.2016.91>
- Reja, V. K., Varghese, K., & Ha, Q. P. (2022). Computer vision-based construction progress monitoring. *Automation in Construction*, 138, 104245. <https://doi.org/10.1016/j.autcon.2022.104245>
- Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6), 1137-1149. <https://doi.org/10.1109/TPAMI.2016.2577031>
- Rice, L., Wong, E., & Kolter, Z. (2020). Overfitting in adversarially robust deep learning. In H. Daumé III & A. Singh (Eds.), *37th International Conference on Machine Learning* (pp. 8093-8104), PMLR.
- Roberts, D., & Golparvar-Fard, M. (2019). End-to-end vision-based detection, tracking and activity analysis of earthmoving equipment filmed at ground level. *Automation in Construction*, 105, 102811. <https://doi.org/10.1016/j.autcon.2019.04.006>
- Roh, S., Aziz, Z., & Peña-Mora, F. (2011). An object based 3D walk-through model for interior construction progress monitoring. *Automation in Construction*, 20(1), 66-75. <https://doi.org/10.1016/j.autcon.2010.07.003>
- Roh, S., Peña-Mora, F., Fard, M. G., & Han, S. (2009). Visualization application for interior progress monitoring in 3D environment. In S. T. Ariaratnam & E. M. Rojas (Eds.), *Construction Research Congress 2009: Building a Sustainable Future* (pp. 51-60). American Society of Civil Engineers. [https://doi.org/10.1061/41020\(339\)6](https://doi.org/10.1061/41020(339)6)
- Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In N. Navab, J. Hornegger, W. M. Wells, & A. F. Frangi (Eds.), *Medical image computing and computer-assisted intervention* (pp. 234-241). Springer. https://doi.org/10.1007/978-3-319-24574-4_28
- Saunders, M., Lewis, P., & Thornhill, A. (2009). *Research methods for business students* (5th ed.). Pearson.
- Shafieezadeh-Abadeh, S., Kuhn, D., & Esfahani, P. M. (2019). Regularization via mass transportation. *Journal of Machine Learning Research*, 20(103), 1-68.

- Shahi, A., Cardona, J. M., Haas, C. T., West, J. S., & Caldwell, G. L. (2012). Activity-based data fusion for automated progress tracking of construction projects. In H. Cai, A. Kandil, M. Hastak, & P. S. Dunston (Eds.), *Construction Research Congress 2012: Construction Challenges in a Flat World* (pp. 838-847). American Society of Civil Engineers.
<https://doi.org/10.1061/9780784412329.085>
- Shahi, A., West, J. S., & Haas, C. T. (2013). Onsite 3D marking for construction activity tracking. *Automation in Construction*, 30, 136-143.
<https://doi.org/10.1016/j.autcon.2012.11.027>
- Shahi, A., Safa, M., Haas, C. T., & West, J. S. (2015). Data fusion process management for automated construction progress estimation. *Journal of Computing in Civil Engineering*, 29(6). [https://doi.org/10.1061/\(asce\)cp.1943-5487.0000436](https://doi.org/10.1061/(asce)cp.1943-5487.0000436)
- Shamsollahi, D., Moselhi, O., & Khorasani, K. (2021). A timely object recognition method for construction using the Mask R-CNN architecture. In F. Chen, L. Thomas, & B. Ioannis (Eds.), *38th International Symposium on Automation and Robotics in Construction (ISARC 2021)* (pp. 372-378). The International Association for Automation and Robotics in Construction.
<https://doi.org/10.22260/ISARC2021/0052>
- Shi, Z., Zhu, M. m., Guo, B., Zhao, M., & Zhang, C. (2018). Nighttime low illumination image enhancement with single image using bright/dark channel prior. *EURASIP Journal on Image and Video Processing*, 2018(1), 13.
<https://doi.org/10.1186/s13640-018-0251-4>
- Shirowzhan, S., Sepasgozar, S., & Liu, C. (2018). Monitoring physical progress of indoor buildings using mobile and terrestrial point clouds. In C. Wang, C. Harper, Y. Lee, R. Harris, & C. Berryman (Eds.), *Construction Research Congress 2018: Construction Information Technology* (pp. 602-611). American Society of Civil Engineers. <https://doi.org/10.1061/9780784481264.059>
- Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1), 1-48. <https://doi.org/10.1186/s40537-019-0197-0>
- Sibi, P., Jones, S. A., & Siddarth, P. (2013). Analysis of different activation functions using back propagation neural networks. *Journal of theoretical and applied information technology*, 47(3), 1264-1268.
- Simonyan, K., & Zisserman, A. (2014). *Very deep convolutional networks for large-scale image recognition*. *arXiv*. <https://doi.org/10.48550/arXiv.1409.1556>
- Slaton, T., Hernandez, C., & Akhavian, R. (2020). Construction activity recognition with convolutional recurrent networks. *Automation in Construction*, 113, 103138. <https://doi.org/10.1016/j.autcon.2020.103138>
- Smith, L. N. (2018). A disciplined approach to neural network hyper-parameters: Part 1--learning rate, batch size, momentum, and weight decay. *arXiv*.
<https://doi.org/10.48550/arXiv.1803.09820>
- Soekhoe, D., Putten, P. v. d., & Plaat, A. (2016). On the impact of data set size in transfer learning using deep neural networks. In H. Boström, A. Knobbe, C. Soares, & P. Papapetrou (Eds.), *Advances in Intelligent Data Analysis XV* (pp. 50-60). Springer. https://doi.org/10.1007/978-3-319-46349-0_5
- Stake, R. E. (2013). *Multiple case study analysis*. Guilford press.
- Sun, C., Shrivastava, A., Singh, S., & Gupta, A. (2017, October 22-29). *Revisiting unreasonable effectiveness of data in deep learning era* [Paper presentation]. 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy.
<https://doi.org/10.1109/ICCV.2017.97>

- Szeliski, R. (2010). *Computer Vision: Algorithms and Applications*. Springer Science & Business Media.
- Tamayama, H., Ito, K., & Nishimura, T. (2002, June 13-15). *Technology trends of high-definition digital still camera systems* [Paper presentation]. 2002 Symposium on VLSI Circuits. Digest of Technical Papers, Honolulu, Hawaii, United States. <https://doi.org/10.1109/VLSIC.2002.1015056>
- Teizer, J. (2015). Status quo and open challenges in vision based sensing and tracking of temporary resources on infrastructure construction sites. *Advanced Engineering Informatics*, 29(2), 225-238. <https://doi.org/10.1016/j.aei.2015.03.006>
- Torrey, L. and Shavlik, J. (2010) Transfer Learning. In E.S. Olivas, J.D.M. Guerrero, M. Martinez-Sober, J. R. Magdalena-Benedito, & A. J. S. López (Eds.), *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques* (pp.242-264). IGI Global. <https://doi.org/10.4018/978-1-60566-766-9.ch011>
- Turkan, Y., Bosche, F., Haas, C. T., & Haas, R. (2012). Automated progress tracking using 4D schedule and 3D sensing technologies. *Automation in Construction*, 22, 414-421. <https://doi.org/10.1016/j.autcon.2011.10.003>
- Turkan, Y., Bosché, F., Haas, C. T., & Haas, R. (2013). Toward Automated Earned Value Tracking Using 3D Imaging Tools. *Journal of Construction Engineering and Management*, 139(4), 423-433. [https://doi.org/10.1061/\(asce\)co.1943-7862.0000629](https://doi.org/10.1061/(asce)co.1943-7862.0000629)
- Vaishnavi, V., & Kuechler, W. (2004). *Design research in information systems*. CRC Press.
- Vaishnavi, V. K. (2007). *Design science research methods and patterns: innovating information and communication technology*. Auerbach Publications.
- Valero, E., Adan, A., & Cerrada, C. (2012). Automatic construction of 3D basic semantic models of inhabited interiors using laser scanners and RFID sensors. *Sensors (Basel)*, 12(5), 5705-5724. <https://doi.org/10.3390/s120505705>
- Valero, E., Adán, A., & Bosché, F. (2016). Semantic 3D reconstruction of furnished interiors using laser scanning and RFID Technology. *Journal of Computing in Civil Engineering*, 30(4). [https://doi.org/10.1061/\(asce\)cp.1943-5487.0000525](https://doi.org/10.1061/(asce)cp.1943-5487.0000525)
- Van Aken, J., Chandrasekaran, A., & Halman, J. (2016). Conducting and publishing design science research: Inaugural essay of the design science department of the Journal of Operations Management. *Journal of Operations Management*, 47, 1-8. <https://doi.org/10.1016/j.jom.2016.06.004>
- Wang, B., Wang, Q., Cheng, J. C. P., & Yin, C. (2022). Object verification based on deep learning point feature comparison for scan-to-BIM. *Automation in Construction*, 142, 104515. <https://doi.org/10.1016/j.autcon.2022.104515>
- Wang, J., Ma, Y., Zhang, L., Gao, R. X., & Wu, D. (2018). Deep learning for smart manufacturing: Methods and applications. *Journal of Manufacturing Systems*, 48, 144-156. <https://doi.org/10.1016/j.jmsy.2018.01.003>
- Wang, Q., & Kim, M.-K. (2019). Applications of 3D point cloud data in the construction industry: A fifteen-year review from 2004 to 2018. *Advanced Engineering Informatics*, 39, 306-319. <https://doi.org/10.1016/j.aei.2019.02.007>
- Wang, Q., Tan, Y., & Mei, Z. (2020). Computational methods of acquisition and processing of 3D point cloud data for construction applications. *Archives of Computational Methods in Engineering*, 27(2), 479-499. <https://doi.org/10.1007/s11831-019-09320-4>

- Wang, Y., Liao, P.-C., Zhang, C., Ren, Y., Sun, X., & Tang, P. (2019). Crowdsourced reliable labeling of safety-rule violations on images of complex construction scenes for advanced vision based workplace safety. *Advanced Engineering Informatics*, 42. <https://doi.org/10.1016/j.aei.2019.101001>
- Wang, Z., Zhang, Q., Yang, B., Wu, T., Lei, K., Zhang, B., & Fang, T. (2021). Vision based framework for automatic progress monitoring of precast walls by using surveillance videos during the construction phase. *Journal of Computing in Civil Engineering*, 35(1). [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000933](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000933)
- Wei, F., Yao, G., Yang, Y., & Sun, Y. (2019). Instance level recognition and quantification for concrete surface bughole based on deep learning. *Automation in Construction*, 107, 102920. <https://doi.org/10.1016/j.autcon.2019.102920>
- Wei, W., Lu, Y., Zhong, T., Li, P., & Liu, B. (2022). Integrated vision based automated progress monitoring of indoor construction using mask region based convolutional neural networks and BIM. *Automation in Construction*, 140, 104327. <https://doi.org/10.1016/j.autcon.2022.104327>
- Wong, J. K. W., Maghrebi, M., Fini, A. A. F., Golestani, M. A. A., Ahmadnia, M., & Er, M. (2022). Development of a refined illumination and reflectance approach for optimal construction site interior image enhancement. *Construction Innovation, ahead-of-print*(ahead-of-print). <https://doi.org/10.1108/CI-02-2022-0044>
- Xi, D., Qin, Y., & Wang, Y. (2020). Vision measurement of gear pitting under different scenes by deep Mask R-CNN. *Sensors*, 20(15), 4298. <https://doi.org/10.3390/s20154298>
- Xiong, X., Adan, A., Akinci, B., & Huber, D. (2013). Automatic creation of semantically rich 3D building models from laser scanner data. *Automation in Construction*, 31, 325-337. <https://doi.org/10.1016/j.autcon.2012.10.006>
- Xu, X., Xu, S., Jin, L., & Song, E. (2011). Characteristic analysis of Otsu threshold and its applications. *Pattern Recognition Letters*, 32(7), 956-961. <https://doi.org/10.1016/j.patrec.2011.01.021>
- Yang, J., Park, M.-W., Vela, P. A., & Golparvar-Fard, M. (2015). Construction performance monitoring via still images, time-lapse photos, and video streams: Now, tomorrow, and the future. *Advanced Engineering Informatics*, 29(2), 211-224. <https://doi.org/10.1016/j.aei.2015.01.011>
- Yang, L., Cheng, J. C. P., & Wang, Q. (2020). Semi automated generation of parametric BIM for steel structures based on terrestrial laser scanning data. *Automation in Construction*, 112. <https://doi.org/10.1016/j.autcon.2019.103037>
- Yin, R. K. (2009). *Case study research: Design and methods* (4th ed.). Sage Publications.
- Ying, H., & Lee, S. (2019). A Mask R-CNN based approach to automatically construct as-is IFC BIM objects from digital images. In M. Al-Hussein (Ed.), *36th International Symposium on Automation and Robotics in Construction (ISARC 2019)* (pp. 764-771). The International Association for Automation and Robotics in Construction. <https://doi.org/10.22260/ISARC2019/0103>
- Zhang, Q., Chang, X., & Bian, S. B. (2020). Vehicle damage detection segmentation algorithm based on improved Mask R-CNN. *IEEE Access*, 8, 6997-7004. <https://doi.org/10.1109/ACCESS.2020.2964055>
- Zhang, X., Bakis, N., Lukins, T. C., Ibrahim, Y. M., Wu, S., Kagioglou, M., Aouad, G., Kaka, A. P., & Trucco, E. (2009). Automating progress measurement of construction projects. *Automation in Construction*, 18(3), 294-301. <https://doi.org/10.1016/j.autcon.2008.09.004>

- Zhao, H., Shi, J., Qi, X., Wang, X., & Jia, J. (2017, July 21-26). *Pyramid scene parsing network* [Paper presentation]. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, Hawaii, United States.
<https://doi.org/10.1109/CVPR.2017.660>
- Zhao, Z.-Q., Zheng, P., Xu, S.-T., & Wu, X. (2019). Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, 30(11), 3212-3232. <https://doi.org/10.1109/tnnls.2018.2876865>
- Zollmann, S., Hoppe, C., Kluckner, S., Poglitsch, C., Bischof, H., & Reitmayr, G. (2014). Augmented reality for construction site monitoring and documentation. *Proceedings of the IEEE*, 102(2), 137-154.
<https://doi.org/10.1109/JPROC.2013.2294314>

Appendix 1 Research ethics application approval

This research followed the research integrity standards and principles as outlined by the University of Technology Sydney in the “Research Ethics and Integrity Policy and the Research Management Policy”. The Ethics Application to the Human Research Ethics Committee (HREC) was approved under ETH20-5459. The collection of construction site data included the images of framing, insulation and drywall installation, 3D construction drawings and models. The confidentiality and anonymity of the collected data was maintained to ensure ethically responsible research. The ethics approval email is presented as follows. Please note that the original thesis title of this study at the time of the ethics approval is shown below.

Neg Risk approval - ETH20-5459

research.ethics@uts.edu.au <research.ethics@uts.edu.au>

To:

- Research Ethics <research.ethics@uts.edu.au>;
- Biyanka Ekanayake <Biyanka.Ekanayake@student.uts.edu.au>;
- Johnny Wong <Johnny.Wong@uts.edu.au>

This is an automated email

Dear Applicant

Re: ETH20-5459 - "Intelligent Onsite Progress Measurement through Image Recognition and Machine Learning Algorithms"

You have declared your research as Negligible Risk and that it **DOES NOT** involve any of the following:

- Establishment of a register or databank for possible use in future research projects
- Collection, transfer and/or banking of human biospecimens⁽¹⁾
- Any significant alteration to routine care or health service provided to participants
- Interventions and therapies, including clinical and non-clinical trials, and innovations
- Targeted recruitment or analysis of data from any of the participant groups listed in Chapter 4 of the National Statement (or where any of these participants are likely to be significantly over-represented in the group being studied) including:
 - Women who are pregnant and the human fetus
 - Children and young people (under 18 years)
 - People in dependent or unequal relationships
 - People highly dependent on medical care who may be unable to give consent
 - People with a cognitive impairment, an intellectual disability, or a mental illness
 - People who may be involved in illegal activities (including those affected)
 - Aboriginal and Torres Strait Islander Peoples
- Collection, use or disclosure of personal information (except where expert opinion is being canvassed with full disclosure, consent and identification for use in the public domain)
- Collection, use or disclosure of health information⁽²⁾
- Collection, use or disclosure of sensitive information⁽³⁾
- Covert observation, active concealment, or planned deception of participants
- Activity that potentially infringes the privacy or professional reputation of participants, providers or organisations (except where expert opinion is being canvassed with full disclosure, consent and identification for use in the public domain)
- Potential for participants to experience harm (e.g. physical, psychological, social, economic and/or legal)

- Direct contact with UTS staff/students, patients, consumers or members of the public (except where expert opinion is being canvassed with full disclosure, consent and identification for use in the public domain)
- Participants who have a pre-existing relationship with the researcher (except where expert opinion is being canvassed with full disclosure, consent and identification for use in the public domain)
- People unable to give free informed consent due to difficulties in understanding the Information Sheet or Consent Form
- People in other countries

The UTS HRECs consider the following to be of negligible risk:

- (1) use of commercial cell lines and/or supply of blood or blood products from Australian Red Cross Lifeblood
- (2) consensus methods (i.e. Delphi)
- (3) use of existing collections of data or records that contain only non-identifiable data about human beings (i.e. no codes)

I declare that I believe this research to be of negligible risk in accordance with the [National Statement on Ethical Conduct in Human Research](#) (Chapter 2.1)

I will notify the UTS Human Research Ethics Committees of any variation to this research that may alter the level of risk associated with it. This research will be undertaken in compliance with the UTS Research Ethics and Integrity Policy or any replacement or amendment thereof.

This research will be undertaken in compliance with the Australian Code for the Responsible Conduct of Research and National Statement on Ethical Conduct in Human Research.

Please keep a copy of your ethics application form and approval letter on file to show you have considered the risks associated with your research. You should consider this your official letter of approval.

To access this application, please [click here](#). A copy of your application form has been attached to this email.

If you have any queries about this approval, please do not hesitate to contact your local research office or Research.Ethics@uts.edu.au.

Kind regards,
UTS HREC Ethics Secretariat

C/- Research Office
University of Technology Sydney
Research.Ethics@uts.edu.au | [Website](#)
PO Box 123 Broadway NSW 2007

Ref: Ethics 2 -Neg Risk approved (c)