



Universidade do Minho Escola de Engenharia

siamese Network with Triplet Los

Nawel Ben Haj Salah

Al for QA/QC in construction. A Siamese Network with Triplet Loss



European Master in Building Information Modelling

The European Master in Building Information Modelling is a joint initiative of:

UNIVERZA V LJUBLJANI





Minho | 2

Minho



Universidade do Minho Escola de Engenharia

Nawel Ben Haj Salah

AI for QA/QC in construction. A Siamese Network with Triplet Loss

Master Dissertation European Master in Building Information Modelling

Work conducted under supervision of: Hélder Manuel da Silva e Sousa Manuel Afonso Parente



October, 2025

AUTHORSHIP RIGHTS AND CONDITIONS OF USE OF THE WORK BY THIRD PARTIES

This is an academic work that can be used by third parties, as long as internationally accepted rules and good practices are respected, particularly in what concerts to author rights and related matters.

Therefore, the present work may be used according to the terms of the license shown below.

If the user needs permission to make use if this work in conditions that are not part of the licensing mentioned below, he/she should contact the author through the RepositóriUM platform of the University of Minho.

License granted to the users of this work



Attribution CC BY

https://creativecommons.org/licenses/by/4.0/

ACKNOWLEDGEMENTS

The journey of the BIM A+ master's program has been truly memorable, a very rich experience in discovery, learning, and research

I am deeply grateful to the entire BIM A+ team and sincerely thankful to my supervisors, Prof. Parente, Prof. de Sousa, and Prof. Lino, for their unlimited support and guidance throughout this work

My appreciation also goes to the industry partner BIMMS for their collaboration. I would like to particularly thank Mr. Bruno Caires and Mr. Luís Lima for providing all the necessary support to accomplish this research.

My heartfelt gratitude also goes to my mother and my brother for supporting me through every challenge I faced. Finally, I want to thank my friends, both those who have always been there for me and those I was fortunate enough to meet during this master's program. You have been a constant source of inspiration to me.

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

Braga / Gui marães, 03/10/2025

RESUMO

Os métodos tradicionais de QA/QC baseiam-se principalmente em inspeções manuais ou visuais, que são demoradas, subjetivas e propensas a erro humano. A automatização do QA/QC através da integração de Inteligência Artificial (IA), particularmente durante a fase de execução, contribui para garantir a conformidade, minimizar repetições de trabalhos e melhorar os resultados dos projetos.

Nesta investigação é feita uma revisão da literatura sobre o estado atual das tecnologias digitais e das aplicações de IA em QA/QC, desenvolve uma metodologia para integrar a IA em QA/QC com especial enfoque na triplet network of the Siamese architecture e aplica essa metodologia num estudo de caso utilizando dados reais de um projeto de construção. O conjunto de dados é constituído por imagens "asbuilt" e imagens de modelos BIM "as-designed", recolhidas através de uma plataforma de visão computacional.

O estudo de caso investiga um modelo de IA (a Siamese network with triplet loss) para comparar a similaridade entre imagens "as-built" e modelos BIM "as-designed". O modelo é treinado e avaliado em conjuntos de dados pequenos e grandes, sendo a sua precisão analisada em termos de capacidade de generalização e potencial de aplicação em tarefas reais de QA/QC. Os resultados demonstram o potencial das arquiteturas baseadas em redes Siamese para automatizar a deteção de inconsistências entre o projeto e a execução, proporcionando uma base para a futura integração da IA na gestão da qualidade em construção.

Palavras chave: as-built, as-designed, IA, QA/QC, Siamese Network, triplet loss

ABSTRACT

Traditional QA/QC methods largely rely on manual or visual inspections, which are time-consuming, subjective, and prone to human error. Automating QA/QC by integrating Artificial Intelligence (AI), particularly during the execution phase, helps ensure compliance, minimize rework, and improve project outcomes.

This research conducts a literature review on the current state of digital technologies and AI applications in QA/QC, develops a methodology for integrating AI into QA/QC with a particular focus on the triplet network of the Siamese architecture, and applies this methodology in a case study using real construction project data. The dataset consists of as-built site imagery and as-designed BIM model imagery collected through a computer vision platform.

The case study investigates an AI model (a Siamese network with triplet loss) to compare the similarity between as-built site imagery and as-designed BIM models. The AI model is trained and evaluated on both small and large datasets, and its accuracy is assessed in terms of generalization ability and potential for real-world QA/QC tasks. Results demonstrate the potential of Siamese-based architectures to automate the detection of inconsistencies between design and execution and provide a foundation for future integration of AI in construction quality management.

Keywords: AI, as-built, as-designed, QA/QC, Siamese Network, triplet loss

TABLE OF CONTENTS

1.	. INTRO	DDUCTION	1
2.	LITER	ATURE REVIEW	4
		A/QC AND DIGITAL TECHNOLOGIES:	
	2.1.1.	Global Research Trends in Digital Technologies for QA/QC	
	2.1.2.	Advanced Digital Technology Integration in QA/QC	
	2.1.3.	Challenges and gaps in Modern Technology Integration in QA/QC	
		A/QC AND AI	
		IMPLEMENTATION IN DIGITAL TECHNOLOGY FOR QA/QC (CASE S	
	11		,
	2.3.1.	Additive Manufacturing & Material Assessment	11
	2.3.2.	Advanced Inspection & Tracking Systems	12
	2.3.3.	Data Analytics & Communication Tools:	
	2.3.4.	Digital Construction Platforms	
	2.3.5.	Real-Time Monitoring & Embedded Systems	
3.	. МЕТН	IODOLOGY:	20
		VERVIEW OF AI, ML, AND DL:	
	3.1.1.	Neural networks:	
	3.1.2.	Convolutional neural network:	
	3.1.3.	Siamese adaptation:	26
	3.2. DI	GITAL TOOL IMPLEMENTATION, OPENSPACE AI FOR REALITY CAP	
		UCTION	
	3.2.1.	Description of the platform:	32
	3.2.1.	Implemented Technologies in OpenSpace AI	
	3.2.2.	Key Functionalities and Benefits of OpenSpace AI in Construction Projects	
	3.3. M	ETHODOLOGY OF BUILDING THE DATASET	
4.		STUDIES:	
4.		JILDING PRESENTATION:	
		ROCESS OF BUILDING THE DATASET:	
	4.2. FF	Simpler Walls with Minimal Features:	
	4.2.1.	Architectural Walls with Key Features:	
	4.2.2.	Complex Walls with Dense MEP Systems.	
	4.2.3.	Structural-Focused Walls	
	4.2.4.	Special Viewpoints and Spatial Constraints	
		AMESE MODEL WORKFLOW:	
	4.3.1.	Setup:	
	4.3.1.	Load Data set:	
	4.3.2.		44
		1.154.00.1102.1105.1.7010	

	4.3.4.	Building the model:	46
	4.3.5.	Training the model:	47
	4.3.6.	Triplet Loss and Accuracy: Evaluating Similarity in Siamese Networks	48
4	4.4. Rl	ESULTS OF SIAMESE ALGORITHM:	50
		Training and validation loss of the model	
	4.4.2.	Accuracy Evaluation:	54
5.	CONC	LUSIONS	57
RE	EFEREN	CES	60
ΑF	PENDI	CES	64
1	APPEND	IX 1: CODE	64
			65

LIST OF FIGURES

Figure 1 – Countries with affiliated articles (Toyin, 2025)	5
Figure 2 – Number of publications from 2011 to 2024 (Toyin, 2025)	5
Figure 3 – Number of publications from 2008 to 2020 introducing AlexNet for image classificat Y, 2021)	
Figure 4 – The relative participation of countries in diffrent application fields (Xu, Y, 2021)	
Figure 5 – Research field: algorithm (left) and applications (right) (Xu, Y, 2021)	10
Figure 6 – Domains of AI, ML, DL and widely used algorithms. (Baduge, S.K. et al.,2022)	21
Figure 7 – Neural network as a black-box featuring the non-linear relationship between the variante input variables and multi-variate responses. (Zupan, 1994)	
Figure 8 – Comparison between the biological and artificial neuron. The circle mimicking the	neuron's
cell body represents simple mathematical procedure that makes one output signal yj from the	set input
signals represented by the multi-variate vector X. (Zupan, 1994)	22
Figure 9 – Common Activation (Transfer) Functions (examples include threshold, sigmoid, ar functions). (Zupan, 1994)	
Figure 10 – ANN Architectures: Single-Layer vs. Multi-Layer Networks (contrasts shallow and	
ANN structures). (Zupan, 1994)	23
Figure 11 – Order of weight correction in backpropagation. (Zupan, 1994)	24
Figure 12 – Schematic of a feedforward neural network and a single neuron. (Baduge, S.K. <i>et a</i>	
Figure 13 – Typical architecture of CNN. (Baduge, S.K. et al.,2022)	26
Figure 14 - Typical network structure of Siamese network (Li, Y., Chen, C.L.P., 2022)	27
Figure 15 – Picked positive and negative image pairs of evaluation datasets. (Melekhov et al. 2	.016) 27
Figure 16 – Model structure. Proposed network architecture (sHybridNet) for image matching. E	3ranches
1 and 2 have the same HybridCNN structure. (Melekhov et al. ,2016)	28
Figure 17 - Comparison of tracking results between Ground Truth, SiamFC, and the proposed	tracker.
(He et al., 2018)	28
Figure 18 – Architecture of the proposed twofold SA-Siam network (He et al., 2018)	29
Figure 19 - The pipeline of the StructSiam algorithm. (Zhang et al., 2017)	29
Figure 20 – Qualitative evaluation of the StructSiam algorithm compared with other state-of-the-	-art real-
time trackers. (Zhang et al., 2017)	30
Figure 21 – Triplet network structure of three branches (anchor, positive, negative) feeding into	distance
comparisons, Hoffer and Ailon (2015)	31
Figure 22 – Example of 360° Reality Capture and Walk Path Using OpenSpace AI (openspace	.ai) 32
Figure 23 – Visual Progress Monitoring with Color-Coded Status per Unit (openspace.ai)	32
Figure 24 – Example of Field Notes and Issue Management Interface (openspace.ai)	33
Figure 25 - Side-by-Side Comparison of Site Reality Capture and BIM Model in Op	enSpace
(openspace.ai)	34
Figure 26 –BIM 3D Model Global View of the Facility	
Figure 27 – BIM 3D Model Perspective of Technical and Equipment Areas	36
Figure 28 – Ground Floor Plan	

Figure 29 – First Floor Plan	37
Figure 30 – Roof Level Plan	38
Figure 31 – Site/BIM pair showing a simple wall with radiator and pipe	39
Figure 32 – Site/BIM pair of a simple wall with a door opening	39
Figure 33 – Site/BIM pair of an architectural wall with door, window, radiator, and pipes	40
Figure 34 – Site/BIM pair of an architectural wall with radiator, pipes, and large glazed openin	gs 40
Figure 35 – Site/BIM pair of an architectural wall with a narrow vertical glazed opening	40
Figure 36 – Site/BIM pair of walls with dense MEP systems (fire protection pipes and valves)	41
Figure 37 – Site/BIM pair of a wall integrating ceiling-level MEP elements (ducts and pipes)	41
Figure 38 – Site/BIM pair of a wall with exposed structural elements (steel columns, diagonal	bracing,
and overhead ducts)	42
Figure 39 – Site/BIM pair of a wall with exposed structural frame (steel columns and diagonal	bracing)
	42
Figure 40 – Site/BIM pair of a narrow corridor with vertical supports, suspended trays, and do	orway43
Figure 41 - Site/BIM pair of a wall with constrained spatial perspective with structural as	nd MEP
integration	43
Figure 42 – Example of displaying three triplets in a 3×3 grid: the anchor image on the left, a	a similar
(positive) image in the middle, and a different (negative) image on the right.	45
Figure 43 –Architecture of the Siamese Network with Triplet Loss	46
Figure 44— Example of Training and validation loss outputs of the model	47
Figure 45 – Sample batch of triplets from the training	47
Figure 46–Accuracy outputs of the model	48
Figure 47– Workflow of Siamese Network Training with Triplet Loss	49
Figure 48 – Graph of training loss and validation loss over the 10 Epochs (before increasing the	dataset)
Figure 49 – sample batch of triplets from the training set before increasing the dataset	52
Figure 50- Graph of training loss and validation loss over the 10 Epochs (after increasing the	dataset)
	53
Figure 51 – sample batch of triplets from the training set after increasing the dataset	54
Figure 52–Triplet Accuracy Evaluation	55
Figure 53– Model evaluation of Triplet Accuracy	56
Figure 54–Triplet Accuracy Evaluation	56

LIST OF TABLES

Table 1 – Key Digital Innovations for QA/QC in Construction Projects (Toyin, 2025)	6
Table 2 – Case studies for Additive Manufacturing & Material Assessment	11
Table 3 – Case studies for Advanced Inspection & Tracking Systems	13
Table 4 – Case studies for Data Analytics & Communication Tools	15
Table 5 – Case studies for Digital Construction Platforms	16
Table 6 – Case studies for Real-Time Monitoring & Embedded Systems	18
Table 7 – Classification accuracy comparison. (Hoffer and Ailon, 2015)	31
Table 8 – Training and validation loss of the model before increasing the dataset	50
Table 9 – Training and validation loss of the model after increasing the dataset	52

LIST OF ACRONYMS AND ABBREVIATIONS

AI Artificial Intelligence ANN Artificial Neural Network

AR Augmented Reality
BEP BIM Execution Plan

BIM Building Information Modelling
BIM2Graph BIM-to-Graph conversion framework
CNN Convolutional Neural Network

DL Deep Learning

FC Fully Connected (Layer)

GAN Generative Adversarial Network

GNN Graph Neural Network

Industry Foundation Classes to vector

Ifc2vec embedding

LSTM Long Short-Term Memory (network)

ML Machine Learning
MLP Multi-Layer Perceptron

MEP Mechanical, Electrical, and Plumbing

NDA Non-Disclosure Agreement

NN Neural Network

OC-NN One-Class Neural Network

QA/QC Quality Assurance / Quality Control

RANSAC Random Sample Consensus

ReLU Rectified Linear Unit
RNN Recurrent Neural Network
SOM Self-Organizing Map
UAV Unmanned Aerial Vehicle

1. INTRODUCTION

The terms *quality assurance* (QA) and *quality control* (QC) are often used interchangeably. Since QC is part of QA, maintaining a clear distinction between the two is not always straightforward. QA is defined as all planned and systematic actions necessary to provide confidence that a structure, system, or component will perform satisfactorily and conform to project requirements. QC, on the other hand, refers to the specific procedures involved in the quality assurance process. These procedures include planning, coordinating, developing, checking, reviewing, and scheduling the work. Quality is achieved when individuals perform their tasks carefully and in accordance with project requirements (O'Brien, 1989). In the construction industry, QA and QC have long faced inefficiencies due to their reliance on manual inspections. As construction demands increase and projects become more technologically advanced, the need for innovative QA/QC methods that enhance precision and efficiency has become urgent (Toyin and Sattineni, 2025).

The main aim of this research is to address inefficiencies in QA/QC processes, particularly during the execution phase. The study seeks to enhance QA/QC by integrating Artificial Intelligence (AI) to automate defect detection. To achieve this, an AI model—a Siamese network with triplet loss—will be implemented to compare as-built site imagery with as-designed BIM models. The model will be trained and evaluated on both small and large datasets, and its accuracy will be assessed in terms of generalization ability and potential for real-world QA/QC tasks. The central focus of this study is therefore the automation of comparing as-built data with as-designed models using AI.

This research is conducted in collaboration with BIMMS, a company specialized in Integrated Engineering Services through Digital Construction. The company uses a platform based on computer vision, which provides access to imagery from drones, 360° cameras, mobile devices, and laser scanners. This platform represents an advanced digital construction tool, as it enables remote visual visits to construction sites and aligns real-world conditions with BIM. It adds significant value to QA/QC by facilitating remote visual inspection and progress monitoring. Nevertheless, even when conducted remotely, visual inspection remains time-consuming and prone to human error. Against this backdrop, the research aims to increase the level of automation in QA/QC tasks, moving beyond exclusively visual methods.

The study will adopt a diverse approach consisting of a literature review, a methodology chapter, and a case study with empirical analysis in order to fulfil its objectives. The literature review will begin by emphasising the limitations of traditional methods and the potential of advanced digital technologies in QA/QC. It will then highlight AI-driven solutions in particular and identify research gaps in existing studies. Finally, the review will classify case studies into categories of digital innovations for QA/QC in construction projects—such as additive manufacturing, advanced inspection and tracking, and data analytics. For each category, the case studies will be summarised by specifying their objectives, the AI tools used, the results obtained, and the identified gaps.

The methodology will address two components of the case study. The first concerns the future choice of the AI model and will therefore introduce the concepts related to AI. The second will focus on the preparation of the dataset, beginning with its origin in a computer vision platform and followed by data processing. This part will describe the techniques applied to improve both the visual quality of the images and the model's ability to detect relevant features.

The chapter will begin by defining concepts related to Artificial Intelligence (AI). It will provide a general introduction to AI and explain its main subfields, namely Machine Learning (ML) and Deep Learning (DL). The discussion will then examine Artificial Neural Networks (ANNs) as a class of ML methods, preparing the ground for an explanation of Convolutional Neural Networks (CNNs). CNNs are deep ANNs designed to process grid-structured data, such as pixel-based images, and they will serve as subnetworks within more advanced models such as the Siamese network.

The following subchapter will focus on the Siamese network, which will constitute the core of the final AI model. It will outline its evolution and applications, leading to the introduction of the triplet network, which belongs to the Siamese family but extends the approach to three inputs. The choice of the Siamese network will be justified by its ability to compare two outputs: the as-designed and as-built data. In practice, these outputs will consist of pairs of images extracted from the computer vision platform: one representing a site image and the other the aligned BIM image.

Subsequently, this chapter will introduce the computer vision platform used in the study. It will explain its functionalities for the construction industry, as well as the technologies implemented, before concluding with a discussion of the tool's limitations, particularly in relation to the QA/QC field. This chapter will conclude by presenting the processing of the extracted data, explaining the selection of images, the challenges encountered, and the techniques applied to frame the images.

The case study chapter will present the practical part of this research, including the obtained results and their analysis. Following this logic, the chapter will begin by introducing the construction project from which the dataset will be prepared, namely a large-scale data center project. It will then provide details of this dataset, justify the choice of image pairs related to the project, and demonstrate the implementation of the data processing described in the methodology chapter.

Subsequently, the chapter will develop the workflow of the AI model step by step. It will begin with an introduction to the model and an explanation of its triplet network design, which aims to learn visual similarity by training on triplets of images: an anchor representing the site image, a positive representing the corresponding image from the BIM model, and a negative generated by shuffling site and BIM model images. The chapter will then detail the setup of the model, explaining the required libraries. Next, the preparation of the data will be described, from resizing images to preprocessing the triplets and visualising them. The workflow will explain the setup of the Siamese model, starting with the creation of the embedding model, which converts images into vectors, and continuing with the Siamese model itself, which calculates the squared distances between the anchor–positive and anchor–negative embeddings. Using the prepared triplet dataset, the model will then be trained with a triplet loss function that encourages the network to bring similar images closer together while pushing dissimilar ones further apart. Finally, the model's accuracy in distinguishing between similar and dissimilar images will be evaluated.

The last section of this chapter will analyse the results. First, the model will be trained on a small dataset, and the training/validation loss will be evaluated over epochs. Second, the dataset will be increased and undergo the same evaluation in order to track possible overfitting. Finally, the accuracy results will be assessed on both the small and the large datasets to measure how often the model makes correct predictions.

2. LITERATURE REVIEW

This literature review follows a structured methodology, starting with a global overview and moving toward a targeted thematic analysis. It begins by framing the field of QA/QC in construction, emphasising the limitations of traditional methods and the potential of advanced technologies, particularly AI-driven solutions. It subsequently identifies research gaps in existing studies. Based on these digital technologies, the literature review classifies case studies into categories such as additive manufacturing, advanced inspection and tracking, and data analytics. For each category, pertinent case studies are summarised by specifying the objectives, the AI tools used, the results obtained, and the identified gaps. Finally, a synthesis highlights the recurring challenges and provides insights for potential enhancements and future research.

2.1. QA/QC and Digital technologies:

Quality Assurance (QA) and Quality Control (QC) are crucial in the construction industry to ensure that projects meet standards for safety, functionality, and durability. The growing complexity and scale of modern projects, combined with the implementation of digital technologies in construction processes, make traditional QA/QC methods—often relying on manual inspections and reactive measures—insufficient to meet the evolving demands of the industry (Tang et al., 2022; Toyin et al., 2025).

Manual inspections in QA/QC are time-consuming and costly in the construction industry. These inefficiencies, combined with increasing construction demands and the growing technological complexity of projects, create an urgent need for innovative QA/QC methods that enhance precision and efficiency. Toyin (2025) presents a study showing that, although attempts have been made to integrate technology into QA/QC practices, gaps remain in understanding precision and efficiency, as well as in documenting and synthesising these advancements.

Traditional methods lack the accuracy and efficiency necessary to manage today's high-risk construction environment (Blinn et al., 2017). Therefore, there is a growing and critical demand for innovative approaches that leverage digital technologies to transform quality management processes (Faybishenko et al., 2022). Emerging technologies such as Non-Destructive Testing (NDT), Building Information Modelling (BIM), Terrestrial Laser Scanning (TLS), and big data analytics are at the forefront of this transformation (Hoegh et al., 2011; Liu and Wen, 2023; Oliveira et al., 2023).

Technological advancements, their applications, and future opportunities show that TLS and BIM enhance inspection accuracy and reduce both time and labour costs, while AI-driven NDT improves the accuracy of defect detection. However, challenges remain, such as the complexity of data processing and difficulties in integration. It is suggested that more precise practical applications of these technologies could make QA/QC more data-driven and efficient. The construction industry can take advantage of these findings to optimise quality management.

2.1.1. Global Research Trends in Digital Technologies for QA/QC

Toyin (2025) suggests that research in this domain has been predominantly driven by contributions from three continents: Asia, North America, and Europe. Figures 1 and 2 show that from 2011 to 2024, China ranks first, with affiliations in eight papers, followed by the USA with six and Germany with five. The United Kingdom contributed three papers, while Singapore and Portugal each had two affiliated papers. Other countries, such as Greece, Sri Lanka, Austria, Slovenia, Switzerland, Sweden, Canada, and Russia, were each associated with only one paper.



Figure 1 – Countries with affiliated articles (Toyin, 2025)

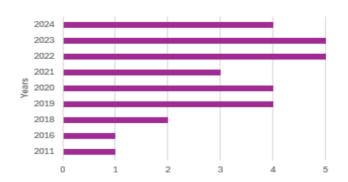


Figure 2 – Number of publications from 2011 to 2024 (Toyin, 2025)

2.1.2. Advanced Digital Technology Integration in QA/QC

The study by Toyin (2025) shows that advanced digital technologies have added value to QA/QC processes in the construction industry. He proposed a classification of these technologies into six groups: Non-Destructive Testing (NDT) Methods, Additive Manufacturing & Material Assessment, Real-Time Monitoring & Embedded Systems, Digital Construction Platforms, Data Analytics & Communication Tools, and Advanced Inspection & Tracking Systems. In addition, he highlighted their practical applications, such as high-precision inspection capabilities, the fabrication of complex components with high accuracy, and the reduction of errors while improving communication among project stakeholders. Table 1 summarises this classification and the contribution of each technology group to QA/QC in the construction field.

Table 1 – Key Digital Innovations for QA/QC in Construction Projects (Toyin, 2025)

Technology Group	Technologies	Application in QA/QC
Technology Group Testing (NDT) Methods	- Ultrasonic Tomography - Terrestrial Laser Scanning (TLS)	These technologies provide non-invasive, high-precision inspection capabilities. Ultrasonic Tomography detects internal anomalies like delamination in concrete structures, ensuring safety and durability. TLS automates geometric quality inspections, capturing 3D point clouds to measure deviations with millimeter-level accuracy, significantly reducing time compared to manual inspections.
Additive Manufacturing & Material Assessment	- 3D Printing - Near-Infrared (NIR) Technology	3D Printing is used for fabricating complex components with high precision, ensuring real-time quality control during the printing process. NIR Technology evaluates critical properties of engineered wood, such as moisture content and surface quality, providing rapid, non-invasive quality assessments. Both technologies focus on improving material quality and ensuring structural integrity.
Real-Time Monitoring & Embedded Systems	- Embedded Sensors - IT-Driven Systems	Embedded Sensors continuously monitor structural and environmental parameters, enabling automated, real-time quality inspections through cloud-based platforms. IT-Driven Systems facilitate efficient information flow and visual management, reducing errors and improving communication among project stakeholders.
Digital Construction Platforms	- BIM - Digital Twin Technology	BIM provides a 3D collaborative environment for integrating project data, supporting proactive quality management and early issue resolution. Digital Twin Technology extends this by offering real-time virtual replicas of physical structures, enhancing predictive maintenance and decision-making. Both technologies streamline quality control and optimize resource allocation.
Data Analytics & Communication Tools	- Big Data & Mobile-ICT - Mobile	Big Data and Mobile-ICT platforms support predictive QA/QC through comprehensive data analysis, improving decision-making and resource optimization. Mobile Apps and 3D Point Clouds enhance field inspections, providing precise site

	Applications & 3D Point Clouds	models and facilitating instant data access and reporting.
Advanced Inspection & Tracking Systems	- LiDAR - Drones, RFID, & Augmented Reality (AR)	LiDAR offers high-accuracy mapping for verifying structural alignments, while Drones capture site images for aerial inspections. RFID tags track material movement, and AR assists in visualizing quality issues and verifying construction tasks interactively. These technologies streamline inspections and improve tracking and verification processes.

2.1.3. Challenges and gaps in Modern Technology Integration in QA/QC

The implementation of advanced digital technologies in QA/QC has led to improvements in inspection accuracy, streamlined workflows, and reductions in labour and time costs. However, the field still faces several challenges, including data processing complexities, interoperability limitations, and high implementation costs. The interpretation of the previous digital technologies classification reveals, for instance, that although drones enhance site inspections, they also present operational and regulatory challenges, limited battery autonomy, dependency on weather conditions, and restricted capabilities for AI-based defect detection. Similarly, while AI-driven defect detection can significantly improve accuracy, it demands substantial computational resources, raises ethical and accountability concerns, and may result in false positives or negatives. These factors contribute to the ongoing challenges of effectively integrating drones and AI into QA/QC processes in the construction industry.

Future research intentions include deploying autonomous inspections, developing AR-BIM-based inspection models, enhancing AI-driven analytics, and implementing quantum-secured data transmission to create self-regulating quality control systems. According to Toyin (2025), some potential directions for future research include:

- Automation and Intelligent Systems: analysing the use of drones equipped with advanced sensors and AI to monitor construction sites in real time, with a focus on the accuracy and efficiency of quality inspections. Developing self-operating robots and drones for real-time quality inspections while minimising human involvement. Creating machine learning algorithms capable of predicting potential quality issues and optimising QA/QC practices through the analysis of historical and real-time data. Guiding future research toward advanced models based on computer vision and AI that can accurately identify and classify defects in construction components using image and video data.
- Advanced Technology Integration: investigating how quantum computing can be used to address the challenge of optimising QA/QC and ensuring the secure transmission of data for records and inspection reports. Leveraging the integration of digital twins into QA/QC

workflows, focusing on how real-time data from digital models can enhance QC processes and support better decision-making. Furthermore, evaluating the potential of AR to assist QA/QC inspections by providing real-time overlays of construction specifications and defect visualisations.

- Fully Autonomous Inspections: QA/QC processes will be conducted by autonomous drones, robots, and smart sensors deployed on construction sites. These systems will continuously inspect quality by comparing real-time data with digital models and project specifications. Construction equipment will have the capacity to automatically correct detected errors using quality monitoring systems. For instance, robotic arms will be able to adjust their own calibration based on sensor feedback to ensure high accuracy and precision in construction processes.
- AI and Machine Learning: analysing vast amounts of data to predict potential quality issues is expected to become increasingly reliable thanks to AI and machine learning. AI-powered image recognition will detect anomalies and non-conformance through video and image analysis.

The following subsection focuses on AI solutions implemented in QA/QC to enhance automation, efficiency, and accuracy—particularly in defect detection, safety monitoring, and predictive quality control—while also highlighting existing research gaps.

2.2. QA/QC and AI

The construction industry is a field where the advancement of Artificial Intelligence (AI) creates new opportunities. AI and machine learning are key areas of innovation in making construction "smart." Applying machine learning in construction offers significant potential, particularly in site supervision, automatic detection, and intelligent maintenance.

A notable milestone in this development was the introduction of AlexNet in 2012, a deep neural network that significantly advanced image classification using deep learning. This sparked a wave of research in computer vision, leading to a steady increase in related publications. As illustrated in Figure 3, the number of papers referencing AlexNet for image classification in construction rose consistently between 2008 and 2020, with a sharp growth trend beginning in 2016 (Xu et al., 2021).

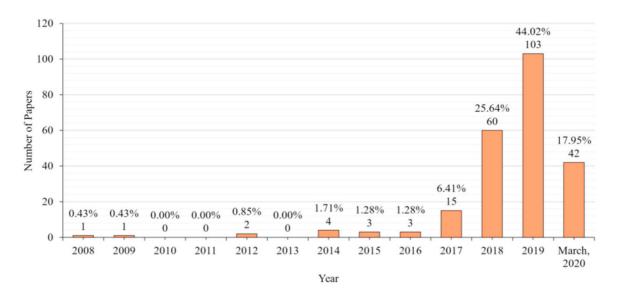


Figure 3 – Number of publications from 2008 to 2020 introducing AlexNet for image classification (Xu, Y, 2021)

Since 2017, the notable performance of deep learning in computer vision has led to its large-scale implementation in various construction fields, such as safety (Fang et al., 2018, 2019; Wu and Cai, 2019), road surveying (Zhang and Yang, 2016; Wu et al., 2019), bridge inspection (Deng et al., 2020; Dorafshan and Azari, 2020; Zhang and Yang, 2020), and on-site operation monitoring (Fang and Li, 2018a; Fang and Ding, 2018; Guo et al., 2020). Research distribution varies geographically. As shown in Figure 4, Chinese researchers predominantly focus on safety management, likely due to the country's massive infrastructure investments—reported to have reached USD 2.6 trillion in 2018, representing 19.6% of national GDP (China National Bureau of Statistics). In contrast, researchers in the United States demonstrate a more diversified focus across QA/QC topics.

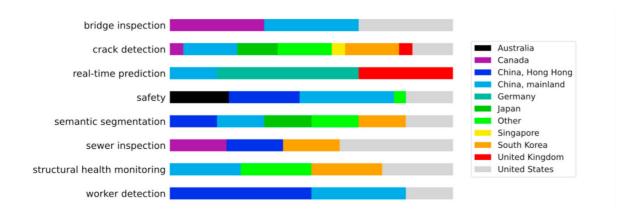


Figure 4 – The relative participation of countries in diffrent application fields (Xu, Y, 2021)

One key observation in the literature is that image data remains the dominant source for applying deep learning in construction. This is largely attributed to the high performance of deep learning in image processing, while other data types are used less frequently. As shown in Figure 5, the most commonly implemented deep learning algorithms are Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and transfer learning approaches. Current research interest is particularly focused on Faster R-CNNs and Long Short-Term Memory (LSTM) networks.

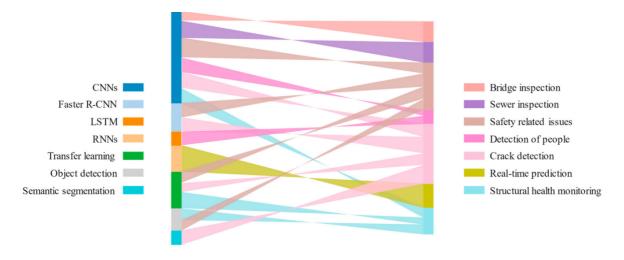


Figure 5 – Research field: algorithm (left) and applications (right) (Xu, Y, 2021)

While the primary implementation of object detection and image segmentation models in construction focuses on defect detection—particularly in concrete crack recognition and bridge inspection—visual characteristics vary across applications. As a result, there is no single algorithm or network structure that is universally applicable. Researchers must therefore select and adapt algorithms and network architectures to address specific construction challenges. In this context, AI has also been applied in various other fields beyond construction. For example, Siamese networks have been used for image matching (Melekhov et al., 2016), real-time tracking (He et al., 2018), and object tracking (Zhang et al., 2017). This type of neural network will be further discussed in the methodology chapter due to its relevance to the present study.

2.3. AI implementation in Digital technology for QA/QC (case studies)

Building on the interpretation provided in the previous section, which outlined the role of digital technologies and AI implementation in QA/QC, this section organises selected case studies according to the previously defined classification of digital technologies implemented for QA/QC. For each category, several AI models will be presented, along with a brief description of their specific applications, strengths, and limitations. This approach not only illustrates the practical implementation of these technologies in real-world construction projects but also highlights the diversity of methods and contexts in which AI is applied to enhance quality assurance and quality control processes.

2.3.1. Additive Manufacturing & Material Assessment

Sri Kalyan (2016) proposes a real-time quality assessment pipeline in Additive Manufacturing (AM). It includes laser scanning to generate point clouds, rasterisation to convert point clouds into height maps, and analysis with a multi-region Hybrid Convolutional Auto-Encoder (HCAE) to detect under-printed and over-printed zones. The study aims to develop an AI-based system capable of detecting and mitigating printing abnormalities in real time and performing in-situ adjustments to enhance the final quality of printed parts. The main challenges to ensuring reliable and repeatable printing are:

- Systematic and random errors
- Lack of real-time monitoring and control
- Difficulty in maintaining optimal process parameters layer by layer

Table 2 – Case studies for Additive Manufacturing & Material Assessment

Technology Group	Refrence	Digital Technology tool	AI implemented	Application in QA/QC
Additive Manufacturing & Material Assessment	(Sri Kalyan, 2016) - AM defect detection	Laser scanning	НСАЕ	Detect print defects

2.3.2. Advanced Inspection & Tracking Systems

Digital technologies play an important role in streamlining inspections and improving tracking and verification processes. For instance, LiDAR enables high-accuracy mapping for verifying structural alignments, while drones capture site images and enable aerial inspections. Additionally, RFID tags track material movement, and AR helps visualise quality issues and interactively verify construction tasks (Toyin, 2025).

(Ahmad W, 2021) analyses the application of supervised machine learning techniques such as Bagging, AdaBoost, Decision Tree, and Gene Expression Programming to predict compressive strength (CS) containing supplementary cementitious materials (SCM) like fly ash and blast furnace slag. To validate the models, the study implemented k-fold cross-validation and sensitivity analysis. Strong predictive capability was shown, with the best performance achieved by the Bagging model. This provides a solution to reduce dependence on time-intensive physical testing in QA/QC in construction. However, the accuracy of ML predictions varies depending on the type of model used, the number and quality of input features, and the data volume. This makes it difficult to generalise models across different datasets.

(Chou, 2022) proposes the Metaheuristics-Optimized Stacking System (MOSS), a predictive framework for estimating scour depth at bridge piers. MOSS combines the Forensic-Based Investigation (FBI) optimisation algorithm with two base models: Least Squares Support Vector Regression (LSSVR) and Radial Basis Function Neural Network (RBFNN) across a stacking ensemble architecture. MOSS achieves significant accuracy gains compared to single ML models, voting ensembles, hybrid methods, empirical equations, and mathematical approaches, by simultaneously optimising all hyperparameters of the constituent models. When tested on laboratory, field, and complex pier datasets, MOSS achieved up to 41% lower prediction errors compared to other approaches. The complexity of scour processes, the limitations of empirical formulas, and the need for adaptable, high-accuracy models across diverse environments are the key challenges addressed.

(De Filippo, 2023) worked on inspections of concrete façades to ensure the safety and sustainability of ageing structures. Traditional approaches, which are mainly qualitative, are time-consuming, expensive, and dependent on human expertise (Kwan & Wong, 2005). Drone integration (UAV) facilitates access to difficult zones and accelerates the collection of visual and thermal data (Mavromatidis et al., 2014). Through both computer vision and deep learning, the collected data enabled automatic pathological defect detection such as cracks, delamination, stains, leakages, debonding, and moisture ingress. This represents a rapid and scalable quantitative approach. Visual detection relied on a model combining RetinaNet with ResNet50 and FPN for macro inspection, while detailed evaluation of defects was realised through a Fully Convolutional Network (FCN-8) with VGG16 for micro inspection. For thermal anomalies, a visual computer approach was implemented with thermal outline detection. However, the authors highlight certain limitations of this study, notably the restricted dataset, false positives related to irrelevant objects, reflections on window glass, and poorly collected data, as well as the necessity to continuously enrich the dataset to improve AI model generalisation.

(The study purpose of Katsamenis, 2023) is to develop an automated, fast, and reliable method to detect road defects in real time from drone images, enhancing safety and reducing inspection and maintenance costs. It implements a YOLOv5 object detection model trained on UAV (drone) images to classify and

detect road defects (cracks and potholes). The images were processed and trained on a GPU, and the model's performance was evaluated using IoU.

(Ma, 2022) proposes an AI system based on deep learning (YOLOv3) to identify painting defects on large steel plates in shipyards. It offers an alternative to traditional visual inspections that rely on workers. The model was trained on around 4,000 images, reached 90% accuracy, and was integrated with a PLC to ensure automatic detection of defects and to stop the conveyor in case of problems. This solution increased productivity by 11% and reduced quality incidents to 1%. It represents a step toward intelligent industry. Future improvements are expected to enhance accuracy through segmentation and data augmentation.

(Raoofi, H. and Motamedi, A., 2020) worked on a deep-learning-based computer vision method to detect and segment excavators and dump trucks on construction sites using Mask R-CNN. To address the challenge of a small dataset, they employed transfer learning from the Microsoft COCO dataset. The model effectively leveraged pre-trained feature extraction, and fine-tuning was applied to enhance accuracy and reduce validation loss. The approach targeted automation of progress monitoring, enabling more efficient and accurate tracking of heavy machinery. Key challenges included the limited size of the training dataset, high variability of jobsite conditions, and the difficulty of balancing segmentation accuracy with processing speed.

Table 3 – Case studies for Advanced Inspection & Tracking Systems

Technology Group	Reference	Digital Technology tool	AI implemented	Application in QA/QC
	(Ahmad W,2021) - ML for concrete CS	-	Bagging, AdaBoost, Decision Tree, GEP	Predict concrete compressive strength
	(Chou, 2022) – Scour depth prediction	-	MOSS (LSSVR + RBFNN + FBI optimizer)	Predict scour depth
Advanced Inspection & Tracking	(De Filippo, 2023) – Façade defects	-	RetinaNet+ResNet50+FPN, FCN-8+VGG16	Detect facade defects
Systems	(Katsamenis, 2023) – UAV road defects	UAV	YOLOv5	Detect cracks/potholes

(Ma, 2022) – Shipyard painting defects	PLC	YOLOv3	Detect painting defects
(Raoofi, 2020) – Machinery detection	-	Mask R-CNN + transfer learning (COCO)	Track machinery
(Sri Kalyan, 2016) – Additive manufacturing (AM) defect detection	Laser scanning	HCAE	Detect print defects

2.3.3. Data Analytics & Communication Tools:

Big Data and Mobile-ICT are technologies that enable predictive QA/QC through comprehensive data analysis, enhancing decision-making processes and improving resource optimisation. Mobile apps and 3D point clouds, on the other hand, provide precise site models and facilitate both reporting and data access.

(Braun, 2019) implements inverse photogrammetry and 4D BIM to automate site image labelling for machine learning applications. UAV-captured images are used to reconstruct 3D point clouds of the construction site, which are then aligned with the BIM model to project semantic data of elements onto 2D corresponding images. This approach enables the automated generation of labelled datasets of high quality for computer vision models. The challenges of this study arise from missing or occluded data—particularly when temporary structures such as scaffolding obscure elements in the scene.

(Braun, 2020) proposes a BIM-integrated machine learning approach to enhance automated construction progress monitoring. By combining Structure from Motion (SfM) for 3D reconstruction with BIM's semantic and temporal knowledge, the method projects as-planned building elements into image space and applies machine learning-based object detection to verify actual progress. This integration enables a detection improvement of up to 50% compared to purely geometric as-planned vs. as-built comparisons, especially in scenarios with occlusions and reconstruction inaccuracies. The approach addresses limitations in current progress monitoring, which often ignores semantic context and suffers from incomplete detection in large, complex sites.

(Serradilla, 2022) presents a review of deep learning models applied to predictive maintenance (PdM). This technology helps identify the most suitable architectures for different steps: anomaly detection, diagnosis, prognosis, and mitigation. The study is based on the analysis of 87 publications that appeared between 2016 and 2021 in scientific databases such as Scopus and IEEE Xplore. It classifies the works according to the PdM stage and the type of architecture used (CNN, RNN, LSTM, autoencoders, GAN, SOM). The main contributions are the complete classification of models, the integration of less-explored

techniques such as SOM, OC-NN, and generative models, and the highlighting of strategies to manage data variability and the lack of failure data, especially through transfer learning and ensemble learning. The study also introduces the often-neglected mitigation phase. The review explains the main limits of this technology: the data are limited, and the models are still complex and not easy to understand. This study is relevant to my dissertation topic because predictive maintenance approaches can be adapted for defect detection and problem anticipation in construction.

Table 4 – Case studies for Data Analytics & Communication Tools

Technology Group	Refrence	Digital Technology A tool	AI implemented	Application in QA/QC
Data Analytics	(Braun, 2019) – BIM photogrammetry labelling	Inverse photogrammetry + 4D BIM	ML dataset generation	Generate training data
& Communication Tools	(Braun, 2020) – BIM and machine learning for construction progress	SfM + BIM	ML detection	Verify progress
	(Serradilla, 2022) – DL model classification for PdM tasks including anomaly detection and mitigation	Predictive maintenance platforms	CNN, RNN, LSTM, Autoencoders, GAN, SOM, OC-NN, Transfer Learning, Ensemble Learning	Classification and comparison of deep learning models for anomaly detection, diagnosis, and mitigation phases of QA/QC workflows

2.3.4. Digital Construction Platforms

These technologies streamline quality control and optimise resource allocation. BIM enables proactive quality management through a 3D collaborative environment, providing early issue resolution, while Digital Twin technology enhances predictive maintenance and decision-making by delivering real-time virtual replicas of physical assets.

(Kayhani, McCabe, and Sankaran, 2023) developed an innovative approach to evaluate the quality of construction elements in complex building-site environments. The purpose of their study is to overcome the limitations of classic methods that directly compare as-designed BIM models with 3D point clouds, which are often affected by noise, occlusions, and partial observability of elements. To achieve this, the authors implemented BIM-GNN, based on a Graph Neural Network (GNN). The methodology relies on three key components: BIM2Graph, which converts the BIM into a graph whose nodes represent objects and whose edges represent their topological and spatial relationships; Ifc2vec, which encodes IFC classes into vectors to better exploit semantic relationships; and BIM-GNN, which classifies each element into one of four quality states: verified, deviated, missing, or no data. The results show a 20–27% improvement in the weighted F1-score compared to conventional approaches and demonstrate the model's ability to infer the state of partially observed or unobserved elements. Nevertheless, this method remains dependent on BIM data quality, requires a minimum amount of labelled data for learning, and can lead to confusion between "missing" and "no data" classes.

(Wei, 2022)'s study presents a deep learning approach implementing image segmentation for the automated progress assessment of walls on a whole floor, with direct integration of results into the BIM environment. The method was tested on a real case in China, particularly on plastering activities. It reached a high segmentation accuracy (mean average precision of 96.8%). With enhancements applied to Mask Region-Based Convolutional Neural Network (Mask R-CNN), its performance was compared with other models.

Table 5 – Case studies for Digital Construction Platforms

Technology Group	Reference	Digital Technology tool	AI implemented	Application in QA/QC
Digital Construction Platforms	(Kayhani, McCabe, and Sankaran, 2023)- BIM-GNN QA	BIM-GNN, BIM2Graph, Ifc2vec	GNN	Classify element quality
	(Wei, 2022) - BIM plastering progress	Mask R-CNN + BIM integration	Mask R-CNN	Track plastering progress

2.3.5. Real-Time Monitoring & Embedded Systems

Embedded sensors continuously monitor structural and environmental parameters, enabling automated, real-time quality inspections through cloud-based platforms. IT-driven systems facilitate efficient information flow and visual management, reducing errors and improving communication among project stakeholders.

For indoor construction progress monitoring, (Ekanayake et al., 2024) focus on automation using the instance segmentation framework with deep learning. To detect and quantify the installation progress of interior drywalls and interior separations, Mask R-CNN was implemented using construction site images. Data was manually labelled for model training. Challenges faced include occlusions and lighting variations, which affected the model's ability to generalise. This study highlights the higher complexity of interior spaces compared with exterior ones.

(Li, J. et al., 2021) propose a computer vision-based productivity evaluation method using CenterNet with a DLA-34 backbone to detect workers and materials during rebar assembly. The system enables automated productivity estimation without disturbing site construction activities. The approach achieves high detection accuracy and shows good consistency with actual site observations.

(Luo, X. et al., 2018) present a method of activity recognition applied to surveillance video footage to generate various and continuous activity labels for each worker appearing in the camera view. The approach uses convolutional neural networks to detect activities from spatial and temporal data flows. A new fusion strategy is proposed to combine the results from both streams. Experimental results show that this method reaches an average accuracy of 80.5%, which is comparable to other activity recognition methods in the field of computer vision, despite challenges caused by significant camera motion, low video resolution, small differences between activity classes, and high variability within the same class. Additionally, the paper demonstrates that this method can support the implementation of efficient and objective work sampling.

Quality control also includes construction progress and quality assessment for specific elements. (Trzeciakiewicz et al., 2025), using image-based automation through on-site camera systems, enabled a drywall analysis. A deep learning-based segmentation model was applied: a modified Mask R-CNN architecture with a ConvNeXt V2 backbone for better feature extraction, additional anchor ratios for narrow objects like metal frames, and deconvolution layers for higher-resolution masks (56×56). The system enabled the detection and classification of various drywall elements, with an analysis module to cluster individual wall segments, estimate camera perspective distortions, and apply corrections. Through this system, it was possible to extract information from images for more accurate progress tracking and quality assessment on construction sites. However, qualitative evaluation relied on visual inspection of segmentation, clustering, and perspective correction on both dataset images and new construction site video frames. The study additionally faced the challenge of a limited dataset size (176 annotated images), which increased the risk of overfitting and poor generalisation.

Construction automation can also rely on computer vision and deep learning for autonomous pick-and-place operations and robotic wall construction. For instance, (Vohra et al., 2021) developed a real-time

visual perception framework for automating brick wall construction using a robotic system. It detects, localises, and estimates the pose of bricks in cluttered environments for autonomous pick-and-place operations through Rotating Box CNN combined with a lightweight 6D pose estimation method based on point cloud analysis and RANSAC. This enabled end-to-end robotic wall construction without manual intervention, focusing on efficiency and precision under limited computational resources.

Table 6 – Case studies for Real-Time Monitoring & Embedded Systems

Technology Group	Reference	Digital Technology tool	AI implemented	Application in QA/QC
Real-Time Monitoring & Embedded Systems	(Ekanayake, 2024) - Indoor progress	Mask R-CNN	Mask R-CNN	Track drywall / separation installation
	Li 2021 - Rebar assembly productivity	CenterNet (DLA-34)	CenterNet	Measure worker productivity
	(Luo, 2018) – Worker activity recognition	-	Two-stream CNN	Monitor worker activities
	Trzeciakiewicz 2025 - Drywall analysis	Modified Mask R-CNN + ConvNeXt V2	Mask R-CNN	Track drywall installation
	Vohra 2021 - Robotic wall construction	Rotating Box CNN + 6D pose estimation	CNN + point cloud	Automate brick placement

Despite the notable progress that QA/QC in construction has witnessed, challenges persist in managing quality amid the increasing demand for large-scale projects. AI, now implemented in various fields, is envisaged for autonomous inspections that can improve inspection accuracy, streamline workflows, and reduce labour and time costs.

However, common challenges that limit the effectiveness of AI-based QA/QC in construction are evident in the reviewed case studies on advanced inspection and tracking systems. These include the complexity of certain processes (such as scour prediction), the limitations of traditional models, and the need for flexible, accurate systems capable of operating across diverse conditions. Issues related to small datasets are observed in UAV-based inspections, along with false detections caused by reflections or irrelevant objects, and the requirement for continuous data enrichment to improve generalisation. Indoor

progress monitoring also faces challenges due to poor lighting, occlusions, and difficulties in generalising models. Worker activity recognition is affected by camera instability, low-resolution footage, and the similarity between different actions. Some studies highlight overfitting risks in drywall analysis due to a limited number of annotated images, while others point out the need for greater computational resources in real-time robotic wall construction.

These challenges highlight the need for better segmentation techniques, data augmentation, and improved model design to increase the accuracy and reliability of AI applications in QA/QC.

3. METHODOLOGY:

This chapter will address two components of the case study. The first concerns the future choice of the AI model and therefore will introduce the concepts related to AI. The second will focus on the preparation of the dataset, beginning with its origin in a computer vision platform and followed by data processing. This part will describe the techniques applied to improve both the visual quality of the images and the model's ability to detect relevant features.

3.1. Overview of AI, ML, and DL:

Human approaches or conventional IT programs do not demonstrate realistic efficiency or accuracy in analysing large amounts of data or in pattern recognition. AI, in contrast, offers the capacity to process massive datasets and recognise patterns through large-scale statistical model building. AI has been implemented since the 1940s; however, it is generally defined as the science that develops intelligent machines or computer programs capable of mimicking human intelligence (Baduge, S.K. et al., 2022).

In recent years, AI has made remarkable progress in several domains such as computer vision, robotics, autonomous vehicles, language translation, gaming, medical diagnosis, speech recognition, and generative design. This progress has largely been achieved thanks to two main technologies: machine learning (ML) and deep learning (DL). ML, a subfield of AI, is used to make predictions and learn useful patterns or representations from datasets. DL, a subfield of ML, can be defined as a learning technique that uses multiple layers of simple and adaptable computing units, commonly known as neural networks.

With the enhancement of computing power, Convolutional Neural Networks (CNNs)—widely used in deep learning—have become one of the most applied techniques in visual object recognition, speech recognition, image and speech synthesis, and machine translation.

Figure 6 illustrates the fields of AI, ML, and DL, as well as widely used algorithms such as MLP (Multi-Layer Perceptron), GAN (Generative Adversarial Network), CNN, RNN (Recurrent Neural Network), LSTM (Long Short-Term Memory Network), and RBFN (Radial Basis Function Network) (Baduge, S.K. et al., 2022).

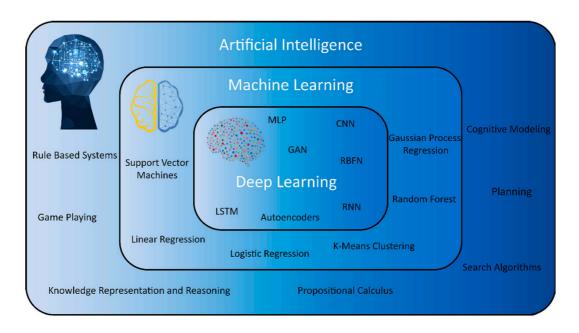


Figure 6 – Domains of AI, ML, DL and widely used algorithms. (Baduge, S.K. et al., 2022)

3.1.1. Neural networks:

Inspired by the structure and function of the human brain, Artificial Neural Networks (ANNs) are a class of machine learning models that have emerged as a family of computational methods. Unlike traditional statistical approaches, ANNs are distinguished by their ability to model complex, non-linear relationships between inputs and outputs. Conceptually, an ANN can be viewed as a black box with multiple inputs and outputs, where information is learned rather than explicitly programmed.

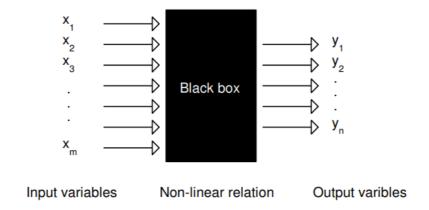


Figure 7 – Neural network as a black box illustrating the non-linear relationship between multivariate input variables and multivariate responses (Zupan, 1994)

The artificial neuron, the basic element of an ANN, is designed to mimic the behaviour of biological neurons. To produce an output, each neuron receives multiple inputs, applies weights, sums them, and passes the result through an activation (transfer) function. In this analogy, the weights correspond to synaptic strengths in biological neurons and represent the knowledge that the network acquires during training.

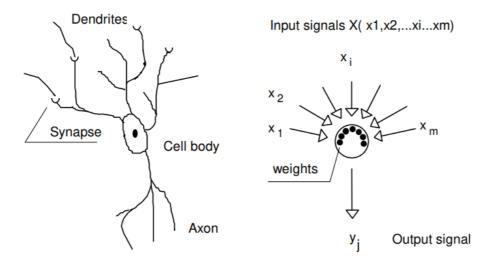


Figure 8 – Comparison between the biological and artificial neuron. The circle mimicking the neuron's cell body represents simple mathematical procedure that makes one output signal yj from the set input signals represented by the multi-variate vector X. (Zupan, 1994)

Nevertheless, the weighted sum of inputs is not the sole determinant of a neuron's output; the activation function also plays a crucial role. Through this non-linear transformation, networks gain the flexibility to capture complex patterns. Common types of activation functions include:

- Threshold function binary output (0 or 1).
- Sigmoid function smooth, continuous output between 0 and 1.
- Radial function localized activation around a central value.

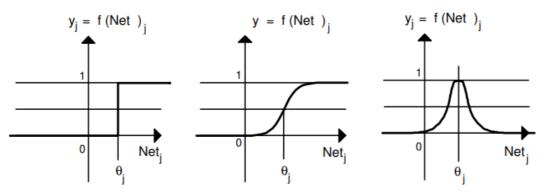


Figure 9 – Common Activation (Transfer) Functions (examples include threshold, sigmoid, and radial functions). (Zupan, 1994)

ANNs are structured in layers:

- Input layer receives the raw variables.
- Hidden layers intermediate processing units that capture patterns.
- Output layer generates predictions or classifications.

Building on this layered structure, multiple hidden layers can be stacked to model increasingly abstract features, forming the foundation of deep learning architectures.

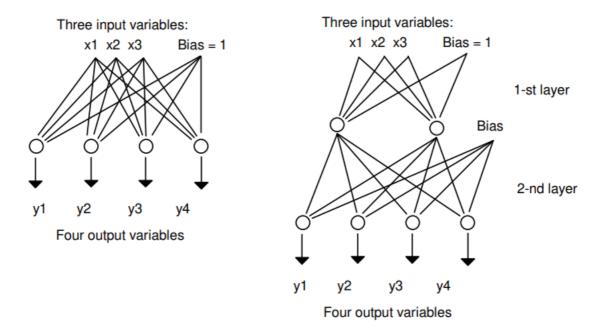


Figure 10 – ANN Architectures: Single-Layer vs. Multi-Layer Networks (contrasts shallow and deeper ANN structures). (Zupan, 1994)

The ability to learn from data by adjusting weights highlights the power of ANNs. According to (Zupan, 1994), there are three major learning paradigms:

- Supervised Learning – Error Backpropagation:

In supervised learning, an ANN is trained with labelled input—output pairs, and error backpropagation is one of the most widely used methods. The term derives from the learning procedure, where the error between predicted and target outputs is propagated backwards to update the weights. The weights of neurons are first corrected in the output layer, then in the second hidden layer, and finally in the first hidden layer.

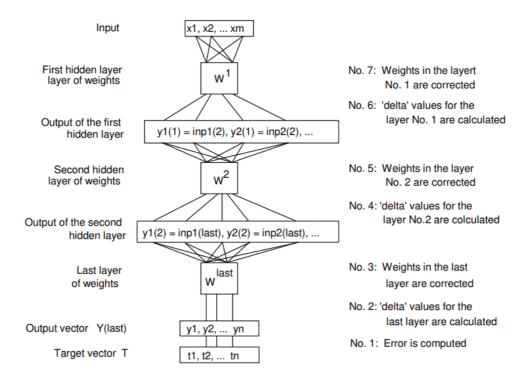


Figure 11 – Order of weight correction in backpropagation. (Zupan, 1994)

- Unsupervised Learning – Kohonen Self-Organizing Maps:

Unsupervised learning does not require labelled outputs, unlike supervised methods, as it discovers the internal structure of the data during the learning process. One well-known example is the Kohonen Self-Organizing Map (SOM). Although less common in modern deep learning, this method represents an important foundation for clustering and visualisation tasks, as it projects high-dimensional data onto a two-dimensional grid while preserving topological relationships.

- Hybrid Models – Counter-Propagation Networks:

Counter-propagation networks are considered a hybrid method, as they combine Kohonen maps with supervised output layers, bridging unsupervised feature mapping with predictive modelling.

Since their discovery, ANNs have formed the foundation of modern deep learning thanks to their non-linear structure. Despite limitations such as computationally intensive training and the risk of overfitting when data is insufficient, it remains essential to understand these classical principles before exploring more advanced architectures. The following section will explain Convolutional Neural Networks (CNNs) as an extension of this paradigm, exploiting spatial hierarchies that make them highly effective for image analysis. Subsequently, Siamese networks will be introduced as an adaptation of ANNs in general, and CNNs in particular, designed for computer vision tasks.

3.1.2. Convolutional neural network:

A Convolutional Neural Network (CNN) is a specialised type of Artificial Neural Network (ANN) designed to process grid-structured data, such as images composed of pixels. CNNs are widely used in image classification and computer vision. They typically include three types of layers: the convolutional layer, the pooling layer, and the fully connected (FC) layer. In conventional CNNs, convolutional layers are followed by pooling layers (or additional convolutional layers such as the ReLU layer), with the FC layer placed at the end, as illustrated in Figure 13.

The input layer contains the raw image data. The core building block of the CNN is the convolutional layer, which uses filters—called kernels or feature detectors—to extract features. A filter is a 2D array of weights smaller than the image size. By sliding the filter across the image, a dot product is calculated between pixel values and filter weights, producing an output array. This process, known as convolution, generates a feature map (also called an activation map or convolved feature).

To significantly reduce memory and computation requirements for large inputs such as images, videos, and audio, CNNs restrict connections to local receptive fields (Figure 13), unlike traditional neural networks (NNs), where all neurons in one layer connect to all neurons in the next (Figure 12). An activation function is then applied after convolution, and the depth of the feature map depends on the number of filters used.

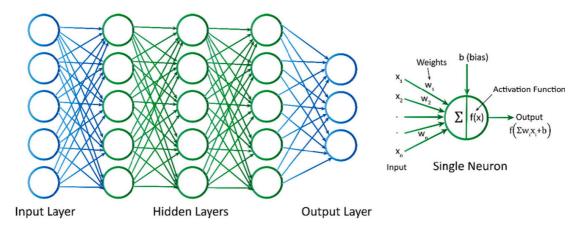


Figure 12 – Schematic of a feedforward neural network and a single neuron. (Baduge, S.K. *et al.*,2022)

The pooling layer is a downsampling operation that reduces the dimensionality of the image by taking either the maximum or the average value within a defined region, thereby retaining important features while reducing computational complexity. The fully connected layer, positioned at the end of the network as shown in Figure 13, contains neurons that are fully connected to the activations of the previous layer in order to perform the classification or final task.

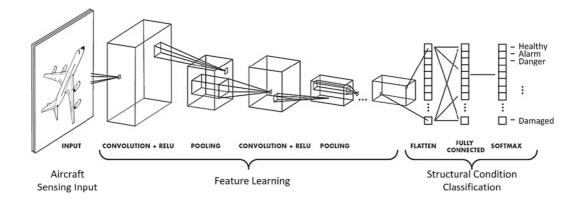


Figure 13 – Typical architecture of CNN. (Baduge, S.K. et al.,2022)

The hyperparameters of CNNs that define the architecture of DL algorithms are the filter size, number of filters, padding, and strides. A cost function is used to train the model, and through backpropagation the filter weights are updated. Different CNN types have been developed by various researchers, including the well-known AlexNet, VGGNet, and ResNet.

3.1.3. Siamese adaptation:

The term *Siamese* refers to the design of a model with two identical subnetworks. A basic Siamese network is composed of two subnetworks—commonly CNNs in computer vision. However, the architecture has since evolved, and it can now consist of two or more identical subnetworks with shared weights. Figure 14 illustrates the typical network structure of a Siamese network. It is designed to learn a similarity function between pairs of inputs. Each branch transforms its input into a feature representation, and the similarity between them is measured using a metric such as Euclidean distance or cross-correlation. The resulting feature embeddings are compared at the final stage to produce a prediction.

The Siamese network was first proposed in 1993 by Bromley et al. for signature verification. Since then, various novel designs of Siamese networks have been introduced. However, significant progress has only been achieved in recent years due to the relatively high computational requirements of deep learning—based structures. Consequently, further research on Siamese networks has been enabled by improvements in computational hardware capability (Li et al., 2022).

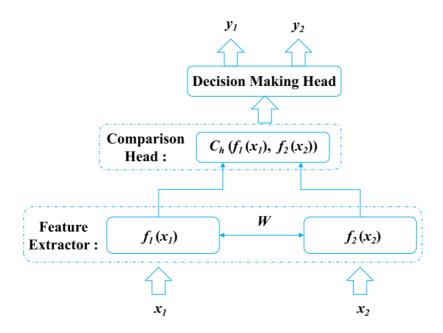


Figure 14 – Typical network structure of Siamese network (Li, Y., Chen, C.L.P., 2022).

Siamese networks are suitable for several purposes, including object tracking, image matching, reidentification, change detection, and product recommendations. Building on this idea, a Siamese network learns by comparing input instances. This architecture has the potential to explore the feature space and obtain discriminative features for downstream tasks. In classification and regression tasks, Siamese networks are widely used to learn effective feature representations for decision-making (Li *et al.*, 2022).

For image matching, (Melekhov *et al.*, 2016) define it as a structure consisting of two identical branches that share weights and parameters (Figure 16). The main goal is to learn optimal feature representations of input pairs, where matched images are pulled closer together and unmatched images are pushed farther apart.



Figure 15 – Picked positive and negative image pairs of evaluation datasets. (Melekhov et al. 2016)

The images in Figure 15 are partially occluded and captured under varying lighting and weather conditions, with differences in viewpoint and appearance. These factors make the generic image-matching task more challenging.

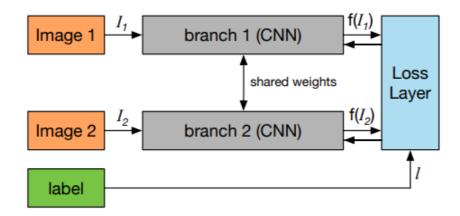


Figure 16 – Model structure. Proposed network architecture (sHybridNet) for image matching. Branches 1 and 2 have the same HybridCNN structure. (Melekhov et al., 2016)

On the other hand, (He et al., 2018) emphasise the conceptual role of Siamese networks in real-time tracking, noting that these architectures learn a general similarity function between the target patch (the object to be tracked) and candidate patches (possible object locations in the next frame). This makes them attractive for tracking due to their speed and robustness. Figure 17 shows that the target object is consistently followed by the proposed tracker, even under variations in shooting angle and scale, where SiamFC fails (adapted from He et al., 2018).



Figure 17 – Comparison of tracking results between Ground Truth, SiamFC, and the proposed tracker. (He et al., 2018)

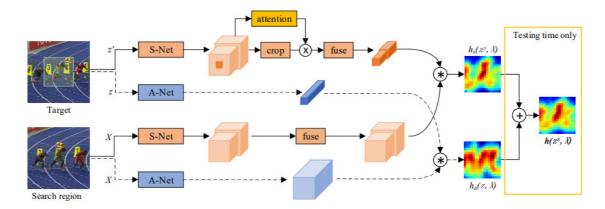


Figure 18 – Architecture of the proposed twofold SA-Siam network (He et al., 2018).

The model combines an appearance network (A-Net) and a semantic network (S-Net), as shown in Figure 18. Features are extracted separately and fused during testing, allowing the tracker to integrate both semantic and appearance information for improved object tracking.

Finally, (Zhang et al., 2017) adapted this function for structured object tracking. Trackers based on Siamese networks select the target from candidate patches using a matching function learned offline on image pairs. This matching function is typically implemented by two-branch CNNs with tied parameters, which take the image pairs as input and predict their similarity.

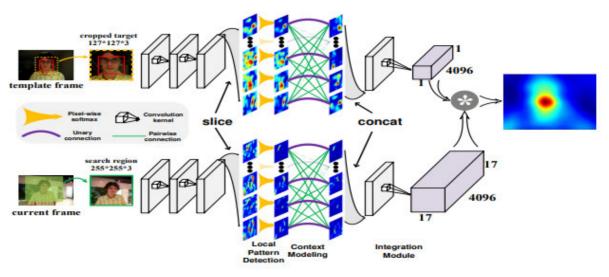


Figure 19 – The pipeline of the StructSiam algorithm. (Zhang et al., 2017)

Figure 19 illustrates that, compared with existing trackers, the results of the StructSiam network demonstrate its ability to handle scale variation, occlusion, and appearance changes. The qualitative evaluation shows that StructSiam produces more focused and stable target localisation, while bounding box comparisons highlight its robustness across challenging video sequences (Zhang et al., 2017).

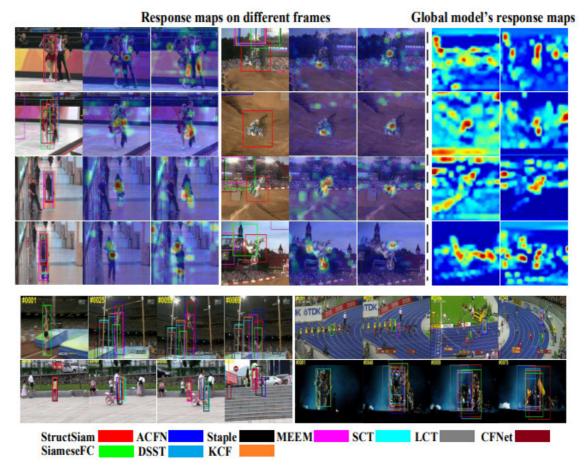


Figure 20 – Qualitative evaluation of the StructSiam algorithm compared with other state-ofthe-art real-time trackers. (Zhang et al., 2017)

Figure 20 shows response maps on different frames (top), illustrating target localisation performance; response maps of the global model (right); and bounding box comparisons across seven benchmark sequences (bottom).

- Triplet Network:

Beyond the conventional Siamese network, which uses identical branches with shared weights to process two inputs, the triplet network represents a more powerful extension designed for metric learning. It belongs to the Siamese family but extends the idea to three inputs: an anchor, a positive, and a negative. (Hoffer and Ailon, 2015) proposed the triplet network for deep metric learning, where x_1 is the anchor input, x_2 is a positive sample, and x_3 is a negative sample (Figure 21). In similarity comparison, x_1 and x_2 are from the same category, while x_3 belongs to a different category. By using both positive and negative pairs simultaneously during training, the triplet network is able to learn more discriminative features.

The triplet network has been applied in image classification, retrieval, re-identification, and other metric learning tasks. Recent studies also suggest its potential for unsupervised extensions, such as temporal or spatial proximity.

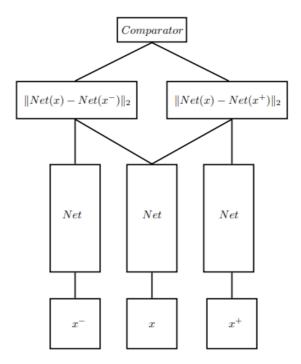


Figure 21 – Triplet network structure of three branches (anchor, positive, negative) feeding into distance comparisons, Hoffer and Ailon (2015)

Table 7 – Classification accuracy comparison. (Hoffer and Ailon, 2015)

Dataset	TripletNet	SiameseNet	Best Known Result (with No Data augmentation)
Mnist	99.54±0.08%	97.9±0.1%	99.61% Mairal et al. (2014); Lee et al. (2014)
Cifar10	87.1%	-	90.22% Lee et al. (2014)
SVHN	95.37%	-	98.18% Lee et al. (2014)
Mnist	99.54±0.08%	97.9±0.1%	99.61% Mairal et al. (2014); Lee et al. (2014)
STL10	70.67%	-	67.9% Lin & Kung (2014)

Table 7 demonstrates that the Triplet Network achieves high classification accuracy across multiple datasets, in some cases outperforming or matching the best-known results without data augmentation. However, it reports Siamese network results only on MNIST. According to (Hoffer and Ailon, 2015), attempts to train Siamese networks on the other datasets did not yield meaningful results, suggesting that the context-dependence of the similarity measure in Siamese networks made training unstable. These findings highlight the robustness of Triplet embeddings in comparison with Siamese networks, which will represent the core of the AI model to be developed in the case study.

3.2. Digital tool implementation, OpenSpace AI for Reality Capture in Construction

3.2.1. Description of the platform:

OpenSpace AI is an integrated platform that enables comprehensive site capture from every angle, creating a shared visual record from pre-construction through operation. It leverages imagery collected from drones, 360° cameras, mobile devices, and laser scanners. The platform automatically pins images to floor plans, aligns real-world conditions with BIM models, and organizes data by sheets and zones, thereby transforming raw imagery into actionable insights for project progress tracking. Its visual documentation features allow users to flag changes, monitor QA/QC through field notes, and facilitate seamless communication between on-site teams and office-based professionals.



Figure 22 – Example of 360° Reality Capture and Walk Path Using OpenSpace AI (openspace.ai)



Figure 23 – Visual Progress Monitoring with Color-Coded Status per Unit (openspace.ai)

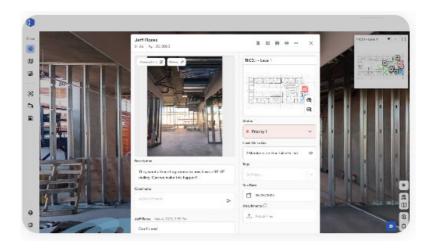


Figure 24 – Example of Field Notes and Issue Management Interface (openspace.ai)

3.2.1. Implemented Technologies in OpenSpace AI

This tool relies on computer vision to automatically align images into an integrated scene, identify and label key features, and map them onto floor plans, thereby providing a visual understanding of the captured environment. It also offers the capability to convert 360° videos into a 3D map by generating a point cloud. This process is achieved by detecting common points between images and estimating the corresponding camera positions during capture.

The platform leverages machine learning to enhance the performance of its AI engine. With each additional site walk, the model learns from the captured data, enabling faster and more accurate image alignment and mapping. Moreover, OpenSpace integrates image-based Simultaneous Localization and Mapping (SLAM)—a technique widely used in robotics, drones, and autonomous vehicles—to estimate the walker's path on a floor plan while simultaneously reconstructing the surrounding environment.

In addition, the system incorporates large language models (LLMs) to interpret and extract insights from reality-capture data, thereby improving automation and decision-making capabilities. Overall, the platform creates a comprehensive visual record and serves as a centralized source of information, allowing field teams to easily view and navigate BIM models on-site. It also provides advanced features such as BIM Compare for visual comparisons, Split View for tracking changes over time, and the ability to transform captured data into actionable insights.

3.2.2. Key Functionalities and Benefits of OpenSpace AI in Construction Projects

This platform provides a wide range of functionalities for different stakeholders in construction projects—including contractors, trades, and project owners—with the aim of improving coordination, documentation, and project oversight throughout the construction lifecycle.

First, it enables remote progress tracking by providing images that clearly document what has been completed and when. This supports the evaluation of material quantities, percentage of completion, and

overall productivity. It also facilitates the annotation of hazards through Field Notes, the generation of detailed reports, and the seamless sharing of information across teams.

Second, the platform enhances (QA/QC) by visually identifying discrepancies before they escalate into significant issues, using tools such as BIM Compare. It also supports project management by enabling professionals to plan construction activities, monitor progress remotely across multiple projects, and reduce the need for frequent site visits, thereby helping mitigate labor shortage risks.

Finally, the platform simplifies project coordination by keeping all captures, conversations, and project activities neatly organized, easily accessible, and securely stored on dedicated servers. It contributes to significant cost savings by reducing travel expenses by nearly 50%, minimizing rework through improved accuracy, and lowering insurance premiums by up to 25%. These benefits are largely achieved through its ability to ensure visual transparency, consistent progress tracking, and reliable documentation throughout the project lifecycle.



Figure 25 – Side-by-Side Comparison of Site Reality Capture and BIM Model in OpenSpace (openspace.ai)

This platform adds significant value to the digitalization of the construction field by integrating cutting-edge AI technologies that enhance project management, including site coordination, progress tracking, quality assurance and quality control (QA/QC), safety monitoring, and cost reduction. However, the tool primarily relies on visual inspection to enable functionalities such as progress monitoring and QA/QC verification. In this research, the aim is to explore approaches for automating QA/QC-related tasks in order to improve efficiency and accuracy.

Through its BIM Compare feature, the platform enabled the collection of a dataset composed of site images and their corresponding BIM model images. Based on this dataset, and by leveraging an AI-powered Siamese network, a model will be developed to automate the comparison process, thereby reducing dependency on manual visual inspection and improving the overall reliability of quality assessment.

3.3. Methodology of building the dataset

To build a consistent and informative dataset for training the Siamese model (Triplet Network), the construction site was explored through a 360-degree virtual walk using the OpenSpace AI software. Walls were chosen as the primary focus for dataset creation, as they were considered simpler objects for testing the network in its initial phase. During the image selection process, several criteria were applied to enhance both the visual quality of the images and the model's ability to detect relevant features.

- Framing: Images were cropped to retain the most relevant part of the wall, ensuring that architectural and MEP elements—such as windows, doors, or pipes—were visible. This allowed the model to recognize and differentiate between various features.
- Angle and Perspective: A consistent perspective was maintained to reduce variation and minimize noise in the dataset. Image extraction was therefore based on a single, fixed viewpoint for both site images and model images, ensuring comparability.
- Lighting Control: Lighting conditions were carefully considered. Images with natural or site lighting that provided clear visibility were prioritized, while strong shadows and overexposed areas were avoided to preserve detail and improve feature recognition.
- Focus and Sharpness: To facilitate accurate detection, only images of fully completed walls were selected, while those under construction were excluded. This improved the model's ability to capture edges, textures, and feature boundaries.

The image collection process was time-consuming, primarily due to occlusion issues, where objects in the foreground blocked relevant background features. On construction sites, occlusion was frequently caused by workers, scaffolding, materials, equipment, or temporary lighting. In such cases, image capture had to be delayed or adjusted until the obstruction was removed.

Another challenge arose in very narrow spaces, where it was not always possible to capture images with appropriate framing or perspective. Furthermore, the dataset was constrained by the 360-degree virtual walk, which did not always cover all the required areas of the site.

For optimal results, it was essential to anticipate how the model would process the dataset, particularly when comparing site images to model images. In this sense, a visually consistent dataset required the establishment of clear and coherent selection criteria to ensure quality and comparability.

4. CASE STUDIES:

4.1. Building presentation:

In compliance with the Non-Disclosure Agreement (NDA), the description of this project includes only general information, without disclosing specific names, locations, or areas. The project concerns a data center in Europe with a capacity of 12 MW, organized into several functional zones with clearly defined circulation routes.



Figure 26 -BIM 3D Model Global View of the Facility



Figure 27 – BIM 3D Model Perspective of Technical and Equipment Areas

The ground floor comprises large plant rooms and areas reserved for future expansion, as well as smaller office and service blocks. Circulation is structured into main routes, secondary access paths, and technical or restricted service zones.

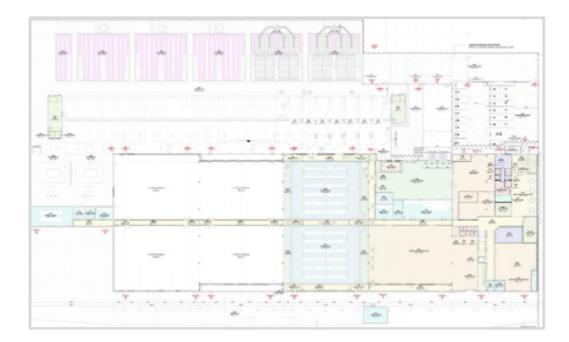


Figure 28 - Ground Floor Plan

The first floor is designed to accommodate administrative and staff functions. It includes office spaces, meeting rooms, a cafeteria, and staff facilities such as restrooms and locker rooms. Circulation is organized horizontally to ensure smooth internal movement between areas, while vertical circulation provides direct links to the ground floor. Both levels are connected through central circulation routes, including stairs, elevators, and corridors.

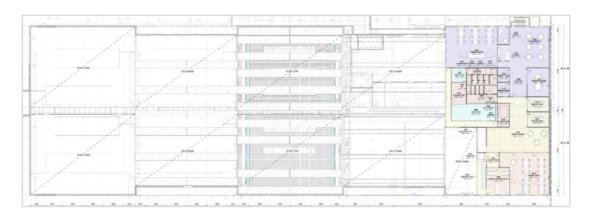


Figure 29 – First Floor Plan

The roof level is dedicated to mechanical and electrical equipment. It includes equipment rooms and technical service areas, with restricted access routes designated for inspection and maintenance activities. This organization ensures functionality and safety while preventing interference with public or administrative functions

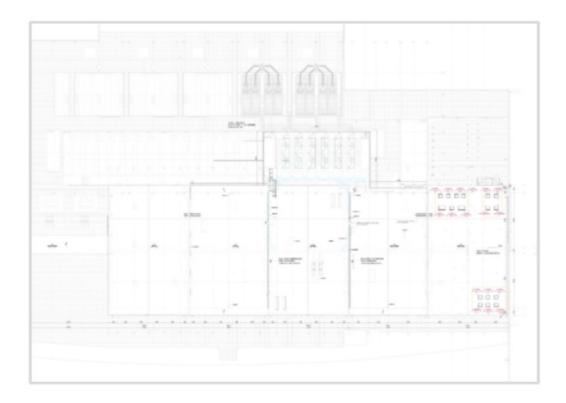


Figure 30 - Roof Level Plan

Overall, the building represents a complex, service-intensive facility, where the coexistence of architectural, structural, and MEP systems creates a challenging yet highly representative environment for dataset collection. It is an under-construction project designed with both architectural spaces and extensive technical installations. Architectural features include glazed openings and doorways, which are recurrent along the walls. Some areas are wide and open, allowing for straightforward frontal viewpoints, while others are narrow corridors or service zones, resulting in restricted perspectives and a higher risk of occlusion. The building also incorporates a dense network of MEP systems—such as fire protection pipes, ventilation ducts, insulated pipes, cable trays, and radiators—distributed along walls, ceilings, and corridors.

4.2. Process of building the dataset:

Taking into consideration the criteria outlined in the methodology chapter for constructing the dataset, pairs of site images and their corresponding BIM model views were selected according to several factors, including angle and perspective, lighting control, and balanced image composition. The predefined camera positions provided by the 360° virtual walk software were used to ensure alignment and comparability through a fixed viewpoint. These characteristics also align with the requirements for building a dataset intended for training an AI model, where architectures such as ResNet or CNN require fixed-size inputs. In addition, walls with varying levels of feature complexity were chosen as the primary focus of this dataset.

The dataset can therefore be classified according to the complexity of wall features, ranging from low-noise baseline cases to high-complexity environments. This classification aims to cover a wide range of construction scenarios and to test the Siamese model's ability to compare site images with BIM model images. The following sections present examples of dataset pairs categorized according to the complexity of features in each group

4.2.1. Simpler Walls with Minimal Features:

At the first level, the image pairs contain simpler features, with only a few elements, such as a radiator and a vertical pipe (Figure 31). The viewpoint is primarily frontal, which ensures better comparability between the site and model images. In some cases, imperfections are visible in the site images due to unfinished elements, such as loose cables (Figure 32). Despite these imperfections, the use of a frontal viewpoint and the limited number of elements result in dataset pairs with reduced noise.

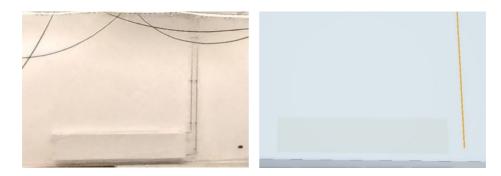


Figure 31 - Site/BIM pair showing a simple wall with radiator and pipe



Figure 32 – Site/BIM pair of a simple wall with a door opening

4.2.2. Architectural Walls with Key Features:

The selected pairs aim to provide sufficient relevant features for comparison while avoiding overcrowding, thereby reducing potential noise. The images were captured at human-eye level, with either a frontal or slightly angled perspective of the wall, for both the site image and the BIM model (Figure 33). The walls include key architectural and MEP features, such as a door, radiator, window opening, and visible cables (Figure 34).



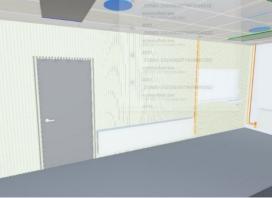


Figure 33 – Site/BIM pair of an architectural wall with door, window, radiator, and pipes



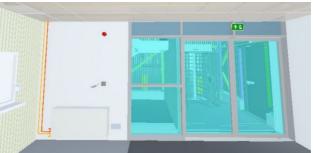


Figure 34 – Site/BIM pair of an architectural wall with radiator, pipes, and large glazed openings

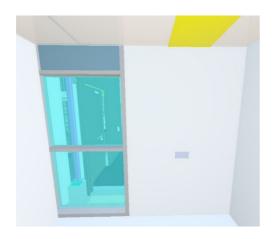




Figure 35 – Site/BIM pair of an architectural wall with a narrow vertical glazed opening

4.2.3. Complex Walls with Dense MEP Systems

Some examples were selected for their higher complexity compared to simpler architectural walls. One pair of images (Figure 36), for instance, shows a larger number of MEP elements, such as fire protection pipes. The viewpoint remains frontal or slightly angled, allowing proper alignment between the site photo and the BIM model. This pair is particularly rich in elements, including pipes, valves, fixtures, and their associated materials and colors. Although such complexity increases the risk of dataset noise, these examples were intentionally included because they provide valuable training material to evaluate how effectively the model can learn to detect and differentiate overlapping and occluded systems.



Figure 36 - Site/BIM pair of walls with dense MEP systems (fire protection pipes and valves)

There are also attempts to integrate ceiling-level MEP elements without altering the primary reference of the dataset, which remains focused on walls. While the viewpoint is still frontal (Figure 37), it extends upward to include components such as ducts and pipes. By incorporating these perspectives, the dataset is enriched and the model's capacity is expanded to learn more about MEP elements. Although these components belong to ceiling zones, they remain visually and functionally integrated with the walls.



Figure 37 – Site/BIM pair of a wall integrating ceiling-level MEP elements (ducts and pipes)

4.2.4. Structural-Focused Walls

Some image pairs are more structurally focused compared to architectural walls, simple walls, or MEP-dense walls. As in most cases, the viewpoint is frontal or slightly angled. These pairs (Figures 38 and 39) emphasize steel structural elements such as vertical columns, diagonal bracing, and overhead ducts. While the site images display real textures and materials, the BIM models represent them with clean surfaces and colored geometries. Although these examples are less cluttered than MEP-heavy pairs, they still provide important structural information that enriches the dataset.

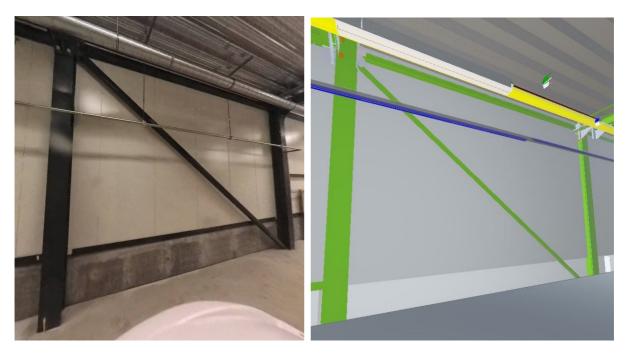


Figure 38 – Site/BIM pair of a wall with exposed structural elements (steel columns, diagonal bracing, and overhead ducts)



Figure 39 – Site/BIM pair of a wall with exposed structural frame (steel columns and diagonal bracing)

4.2.5. Special Viewpoints and Spatial Constraints

Unlike the frontal viewpoints of previous examples, this pair adopts a central "direction of view" perspective. The images (Figures 40 and 41) combine architectural constraints—such as the corridor's narrowness and door opening—with technical systems including vertical supports and suspended trays or cable racks.

As in previous cases, the site image appears more cluttered, whereas the BIM model simplifies these elements and provides greater clarity in the spatial arrangement. Such pairs with alternative viewpoints are an important addition to the dataset, as they test the model's ability to handle restricted perspectives and denser vertical elements, despite the increased risk of occlusion.



Figure 40 – Site/BIM pair of a narrow corridor with vertical supports, suspended trays, and doorway



Figure 41 – Site/BIM pair of a wall with constrained spatial perspective with structural and MEP integration

4.3. Siamese model workflow:

This Siamese model is designed to learn visual similarity by training on triplets of images: an anchor (the site image), a positive (the corresponding BIM model image), and a negative (generated by shuffling site and BIM model images).

The workflow begins with setting up the necessary libraries and preparing the dataset. Images are resized to 200×200 pixels and transformed into tensors to make them readable by machine-learning algorithms. The model then ensures the correct pairing of anchor, positive, and negative images, and visualises sample triplets.

Using a CNN-based embedding model, each image is converted into a vector representation, enabling the Siamese network to measure similarity. The differences between embeddings are computed and used to train the model. Training relies on a triplet loss function that encourages the network to bring similar images closer together while pushing dissimilar ones further apart.

The training process uses batching (processing subsets of images) and epochs (full passes over the dataset). Both training and validation loss are monitored to evaluate learning performance and detect possible overfitting. The following section will explain the detailed structure step by step and will be summarised in Figure 47.

4.3.1. Setup:

As a first step, the environment was set up by importing the required libraries and defining the target image size. This step is essential for convolutional models, which require uniform input dimensions. Libraries such as matplotlib.pyplot were imported for plotting images and graphs and used here to display image samples. NumPy handled numerical operations and array manipulation, while os and random supported file and data management. The deep learning framework TensorFlow was implemented as the main library to build and train the model. Path from pathlib ensured clear and consistent file path handling. Several Keras modules were also imported: applications for pre-trained models and architectures, layers for building network components (e.g., Conv2D, Dense), losses for built-in loss functions, ops for backend tensor operations, optimizers (e.g., Adam, SGD) for updating model weights, metrics for performance tracking, Model as the base class for creating custom models, and resnet for implementing the ResNet architecture from Keras Applications. Finally, a fixed input size of 200 × 200 pixels was defined, to which all images were resized.

4.3.2. Load Data set:

Loading the dataset consists of mounting Google Drive and downloading ZIP files using gdown, a command-line tool suitable for the Colab environment. The dataset files include left.zip for anchor images and right.zip for positive images. To allow direct access to these files (images in this case), the process begins by importing the necessary module to interact with Google Drive and mounting it at the appropriate path (/content/drive), with Path from pathlib ensuring clean path handling. This process results in organized image pairs ready for model training.

4.3.3. Preparing the Data:

This part of the model defines the preprocessing pipeline for images used in the Siamese network. Through a sequence of functions, the model:

preprocess_image function: Reads an image file, decodes it, normalizes the pixel values, and resizes it to 200×200 pixels.

preprocess_triplets function: Ensures that all inputs are in a consistent format before being passed to the model. An image tensor represents image data (pixels) as numbers that a machine learning model can process. This function takes the file paths of an anchor, a positive, and a negative image, and returns their corresponding preprocessed image tensors.

Dataset preparation: The model loads and sorts anchor and positive image paths to ensure correct pairing (anchors and corresponding positive images are stored in Drive folders with the same names). It then shuffles and combines these lists to create negative samples.

Visualization step: As a final part of data preparation, a function displays three triplets in a 3×3 grid, each consisting of an anchor image, a similar (positive) image, and a different (negative) image. This allows verification that image pairs are correctly aligned and labeled before training the model (Figure 42).



Figure 42 – Example of displaying three triplets in a 3×3 grid: the anchor image on the left, a similar (positive) image in the middle, and a different (negative) image on the right.

4.3.4. Building the model:

The model converts each image into a vector representation for similarity comparison within the Siamese network. A pre-trained ResNet50, already trained on a large dataset, is used to extract useful features from the images. The embedding model incorporates custom dense layers that generate a 256-dimensional embedding vector. To improve training efficiency and reduce the risk of overfitting, only the deeper layers of ResNet (starting from *conv5_block1_out*) are made trainable, while the earlier layers remain frozen. This setup transforms images into a numerical form for similarity checking, optimising both training speed and accuracy.

The Siamese model is implemented as a custom Keras model that manages both training and evaluation using triplet loss. It is built by defining three input images—anchor, positive, and negative—and passing them through the shared embedding model. A custom *DistanceLayer* then calculates the squared distances between the anchor—positive and anchor—negative embeddings. By applying a margin-based triplet loss, the model ensures that similar images are positioned closer together in the embedding space than dissimilar ones. Custom training and testing steps are also included to manually compute gradients and update the model's weights during training.

$$L(A, P, N) = \max(\|f(A) - f(P)\|^2 - \|f(A) - f(N)\|^2 + \text{margin}, 0)^1$$

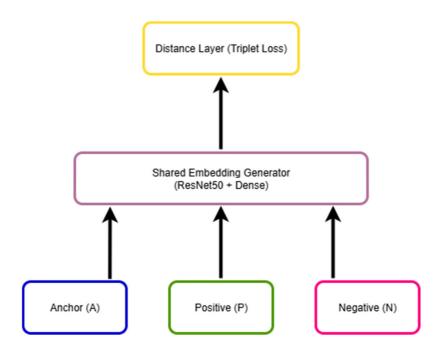


Figure 43 - Architecture of the Siamese Network with Triplet Loss

-

^{1 (}Schroff, F et al, 2015)

4.3.5. Training the model:

Using the prepared triplet dataset, the model is trained for 10 epochs. Validation data is used to monitor performance and mitigate the risk of overfitting. To verify that the data is correctly structured, the model also visualises anchor, positive, and negative images by retrieving a sample batch of triplets from the training set. This visualisation step is essential before relying on the training results, as it confirms that the images are properly aligned and accurately labelled.

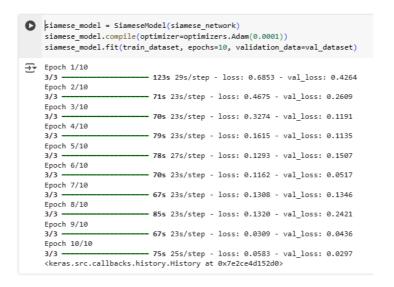


Figure 44- Example of Training and validation loss outputs of the model

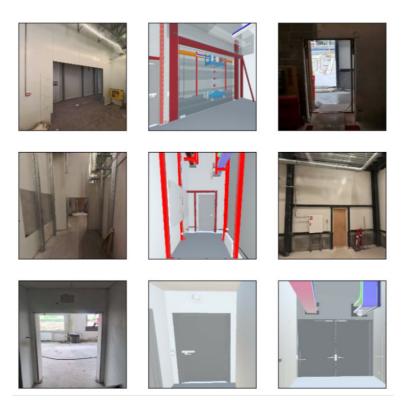


Figure 45 – Sample batch of triplets from the training

4.3.6. Triplet Loss and Accuracy: Evaluating Similarity in Siamese Networks

After training, model performance is evaluated using a metric called triplet accuracy, which reflects how often this condition is satisfied:

Accuracy = Total number of predictions / Number of correct predictions

In this context, a prediction is considered correct if the model ranks the positive image as more similar to the anchor than the negative image, according to the distances between their embeddings. This aligns with the standard definition of accuracy used in classification tasks, but here it is applied to similarity ranking rather than label prediction.

By computing both the loss during training and the accuracy on validation data, we can observe how well the model generalizes to unseen image pairs. This dual evaluation — using distance-based loss and ranking-based accuracy — provides a complete picture of the Siamese model's ability to distinguish visual similarity.

```
[ ] train_acc = compute_triplet_accuracy(train_dataset)
    val_acc = compute_triplet_accuracy(val_dataset)

print("Train accuracy:", train_acc)
print("Validation accuracy:", val_acc)

Train accuracy: 0.9753086419753086
Validation accuracy: 0.9
```

Figure 46-Accuracy outputs of the model

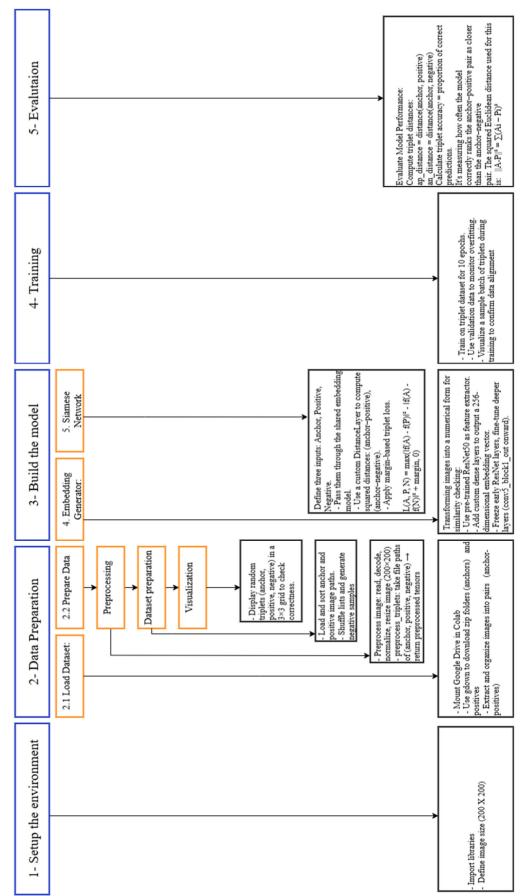


Figure 47- Workflow of Siamese Network Training with Triplet Loss

4.4. Results of Siamese algorithm:

The model is first trained on a dataset of 50 triplets of images, consisting of 50 anchors (site images), 50 positive images (BIM model), and 50 shuffled anchor images used as negatives. Training is performed for 10 epochs, during which both training loss and validation loss are recorded. A sample batch of triplets from the training set is also retrieved for visualisation to confirm correct alignment and labelling.

Both training and validation loss are monitored to evaluate learning performance and to detect possible overfitting. Following this procedure, the model is trained a second time after increasing the dataset to 100 triplets.

This section presents and discusses the results of training and validation loss for both runs (before and after increasing the dataset), followed by an analysis of the accuracy results obtained.

4.4.1. Training and validation loss of the model

Dataset of 50 triplets of images:

Table 8 – Training and validation loss of the model before increasing the dataset

Epoch	Training loss	Validation loss
Epoch 1/10	0.7447	0.4439
Epoch 2/10	0.8525	0.3218
Epoch 3/10	0.3256	0.358
Epoch 4/10	0.325	0.1825
Epoch 5/10	0.198	0.0418
Epoch 6/10	0.2494	0.2508
Epoch 7/10	0.1566	0.0523
Epoch 8/10	0.0849	0.0529
Epoch 9/10	0.0906	0.1311
Epoch 10/10	0.2446	0.0667

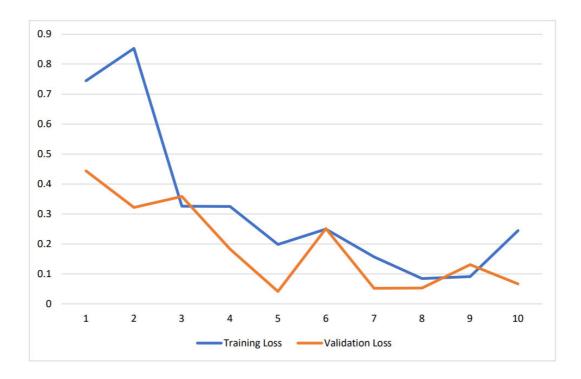


Figure 48 – Graph of training loss and validation loss over the 10 Epochs (before increasing the dataset)

- The training loss starts at approximately 0.75, peaks at epoch 2 with a value of \sim 0.85, and then decreases sharply to \sim 0.3 by epoch 3. This early fluctuation suggests some instability in the training process.
- After epoch 3, the training loss decreases smoothly, stabilising between 0.1–0.2, with a slight increase observed at epoch 10.
- In contrast, the validation loss begins at ~0.45 and decreases steadily, approaching zero by epoch 5. However, at epoch 6, a noticeable spike occurs, indicating fluctuations in validation behaviour.
- This inconsistency in the validation curve suggests that the model has not yet achieved a stable or fully generalisable representation.
- Since the training loss continues to decline while the validation loss occasionally spikes, there remains a significant risk of overfitting.

Overall, the model demonstrates rapid learning but unstable validation performance, strongly suggesting that the small dataset size is a limiting factor. To address this, the model was subsequently trained on an expanded dataset of 100 triplets.

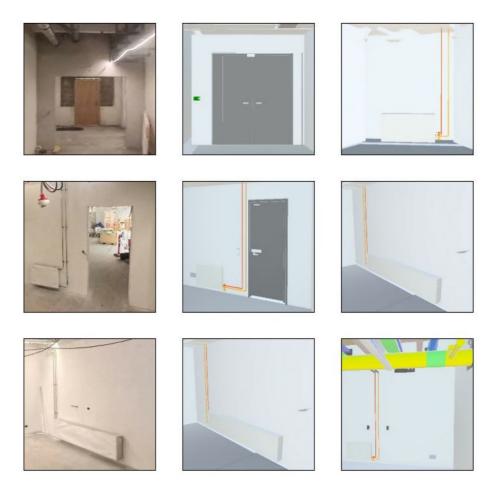


Figure 49 – sample batch of triplets from the training set before increasing the dataset

Dataset of 100 triplets of images:

Table 9 – Training and validation loss of the model after increasing the dataset

Training loss	Validation loss
0.6853	0.4264
0.4675	0.2609
0.3274	0.1191
0.1615	0.1135
0.1293	0.1507
0.1162	0.0517
0.1308	0.1346
0.1320	0.2421
	0.6853 0.4675 0.3274 0.1615 0.1293 0.1162 0.1308

Epoch 9/10	0.0309	0.0436	
Epoch 10/10	0.0583	0.0297	

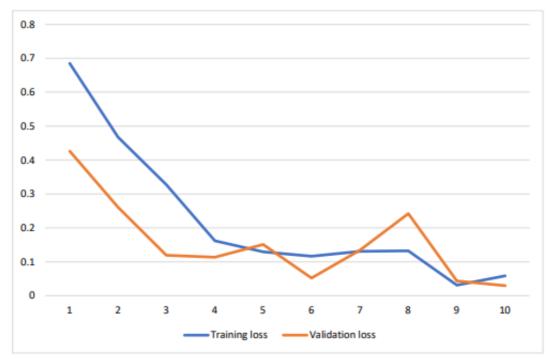


Figure 50– Graph of training loss and validation loss over the 10 Epochs (after increasing the dataset)

As a general observation of this graph, it is encouraging that both the training loss and validation loss decrease sharply during the first few epochs, indicating that the model learns quickly at the start:

- Between epoch 1 and epoch 4, both training and validation loss drop significantly, suggesting that the model is rapidly learning to distinguish between image pairs.
- Some instability is noticeable in the validation set after epoch 4, particularly with the spike at epoch 8. Possible causes include the small size of the validation set, noisy data, or a slightly high learning rate.
- By epoch 10, both losses are again very low (training loss ≈ 0.058 , validation loss ≈ 0.030), indicating that the model has fit the data well without clear signs of overfitting.
- Typically, overfitting occurs when training loss decreases while validation loss increases. In this case, both losses follow the same trend, suggesting that the model is generalizable.

In summary, the model shows rapid convergence in the early epochs (1–4), a fluctuation in validation loss at epoch 8, and low final loss values by epoch 10. These results indicate that the model is likely generalisable, with no strong evidence of overfitting.

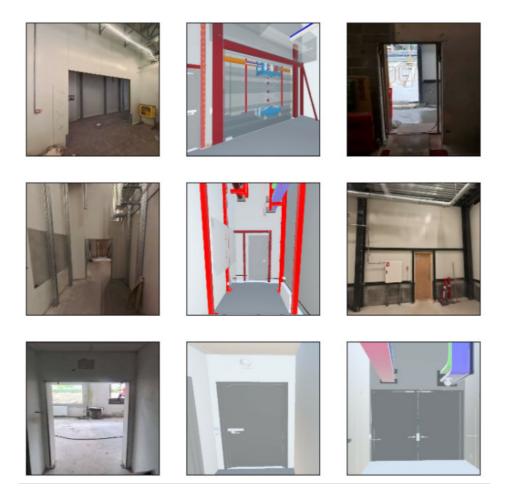


Figure 51 – sample batch of triplets from the training set after increasing the dataset

4.4.2. Accuracy Evaluation:

Triplet accuracy measures how often the model makes a correct prediction of the following condition:

distance (anchor, positive) < distance (anchor, negative)

This means that triplet accuracy represents the percentage of cases where the model successfully learns to bring similar images closer together and push different images farther apart.

The following section presents the triplet accuracy results and their interpretation.

a. Dataset of 50 triplets of images:

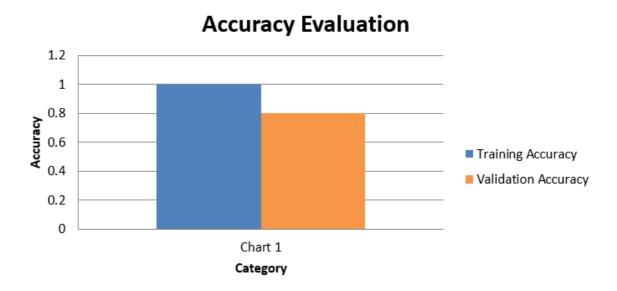


Figure 52-Triplet Accuracy Evaluation

The observed results show a train accuracy of 1.0 (100%) but a validation accuracy of 0.8 (80%). The value of training accuracy suggests that the model has essentially memorized the training triplets. However, the gap with the validation accuracy indicates that the model learned the training examples perfectly, but doesn't generalize equally well to unseen validation triplets. This gap might be a reason of the small dataset, which could lead to a small validation set that is not fully representative.

As a conclusion, with this dataset, the training accuracy reached 100% which suggests that the model memorized the small training dataset. However, the validation accuracy drops to 80%. This gap might highlight overfitting and limited generalization capacity due to the small dataset size.

b. Dataset of 100 triplets of images:

```
[ ] train_acc = compute_triplet_accuracy(train_dataset)
    val_acc = compute_triplet_accuracy(val_dataset)

print("Train accuracy:", train_acc)
print("Validation accuracy:", val_acc)

Train accuracy: 0.9753086419753086
Validation accuracy: 0.9
```

Figure 53- Model evaluation of Triplet Accuracy

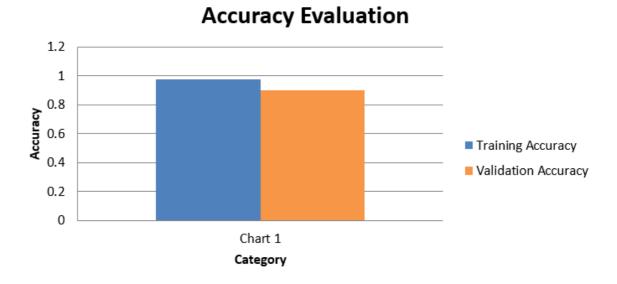


Figure 54-Triplet Accuracy Evaluation

Triplet accuracy measures how often the model correctly ranks the anchor–positive pair as closer than the anchor–negative pair.

The results show:

- Training Accuracy $\approx 97.5\%$ On the training data meaning that 97.5% of triplets were ranked correctly.
- Validation Accuracy \approx 90% On validation data (unseen during training) meaning that 90% of triplets were ranked correctly.

These values indicate that the model has learned the similarity function effectively, achieving high accuracy. The slight drop from training to validation accuracy is normal and even desirable, as it suggests that the model generalizes reasonably well to new, unseen data.

5. CONCLUSIONS

The aim of this work was to contribute to the digitalization of QA/QC in construction through the implementation of an AI-driven solution. The study pursued three main objectives: first, to conduct a literature review on the current state of digital technologies and AI applications in QA/QC; second, to develop a methodology for applying AI to QA/QC, with particular focus on the triplet network within the Siamese architecture; and third, to test this methodology in a case study by training the Siamese model on a real dataset of as-built site imagery and as-designed BIM model imagery, followed by an analysis of the results and an evaluation of its performance.

The literature review has described the limitations of traditional methods in QA/QC, explaining that they are costly, time-consuming, and insufficient for modern complex projects. It then emphasised how advanced digital technologies improve inspection accuracy and efficiency, while also outlining their current limitations, suggesting that more precise practical applications of these technologies could make QA/QC more data-driven and efficient.

This part of the research highlighted how AI-driven solutions, particularly deep learning, can add automation, efficiency, and accuracy to QA/QC, especially in defect detection, safety monitoring, and predictive quality control. It also underlined existing research gaps consisting of dataset limitations, integration problems, false positives, computational constraints, and generalisation issues across studies.

Finally, this review enriched the discussion by presenting case studies of AI-driven technologies for QA/QC, classifying them into technology groups such as Additive Manufacturing & Material Assessment, Advanced Inspection & Tracking, Data Analytics & Communication Tools, Digital Construction Platforms, and Real-Time Monitoring & Embedded Systems.

Following the literature review, this work developed a methodology aimed at implementing an AI-driven solution for a quality control task, specifically comparing as-built with as-designed data. To prepare for the application of the AI-driven solution, this chapter first introduced the concepts of AI, beginning with a general description and then branching into Machine Learning (ML) and Deep Learning (DL).

This explanation established the foundation for Artificial Neural Networks (ANNs) as the basis of modern DL, before moving on to Convolutional Neural Networks (CNNs). The methodology then detailed the Siamese network, which uses CNNs as subnetworks, and concluded with the triplet network, a variant of the Siamese family that represents the core of the AI model developed in this study.

Additionally, the methodology introduced the digital tool implemented for this case study: a platform that combines reality capture (360° imagery, drone data) with AI-driven spatial computing. The BIM comparison feature of this tool enabled the preparation of a dataset. The methodology also described the image selection process, focusing on walls, addressing challenges such as occlusions and perspective limitations, and applying techniques such as lighting control and framing.

The case study chapter presented the practical part of this work. It began by describing the large-scale data center project, which included architectural, structural, and MEP systems—an environment that is both complex and representative for dataset collection.

Pairs of site images and corresponding BIM model views were selected according to criteria such as angle, perspective, lighting control, and balanced image composition. These pairs were used to build a dataset for training the model.

The Siamese model workflow was explained step by step: setting up the libraries, resizing and preprocessing the data, preparing triplets, creating an embedding model, and applying the triplet loss function. The model was trained first on a small dataset, then retrained on a larger one, with both training and validation losses monitored.

The results of implementing the Siamese network with triplet loss were promising. The model learned to bring similar site images and BIM images closer together while pushing dissimilar ones farther apart. Despite the limited dataset, the model showed rapid learning, a decreasing loss trend, and no clear signs of overfitting.

It can therefore be said that the model successfully automated the task of visual similarity checking, demonstrating the feasibility of comparing as-built images with as-designed BIM references. This supports automation in QA/QC, allowing non-compliant or defective elements to be flagged early, reducing human error, and ensuring more consistent inspections.

This study focused on one project element—the wall—although some MEP components were included. A larger scope would require datasets covering other elements, such as ceilings and structural systems. For instance, in data centers, dense ceiling-level MEP installations (ducts, pipes, trays) would need their own dedicated datasets to achieve reliable automation.

Another limitation was dataset size and precision. Time constraints limited the dataset preparation, which impacted the model's generalisation. Small datasets also increased the risk of overfitting and unstable validation performance.

Several extensions of this research are possible.

- Expanded Datasets: Building larger, more precise datasets for different elements (e.g., ceilings, ducts, or structural frames) would improve generalisation.
- Change Detection Methods: Inspired by other fields, such as Wu et al. (2018) in book cover comparison, change detection could be applied in construction to detect discrepancies between BIM models and site images more efficiently.
- Enhanced AI Architectures: Exploring other architectures, such as transformers or self-supervised learning methods, could strengthen performance when labeled datasets are limited.
- Practical Integration: Future studies should focus on integrating such models directly into construction workflows, ensuring that visual inspections, BIM comparisons, and defect detections are part of daily site operations.

This dissertation shows that AI-driven approaches—specifically Siamese networks with triplet loss—can play a meaningful role in advancing the digitalization of QA/QC in construction. While the scope of the study was limited, the findings highlight the potential of combining BIM, computer vision, and deep learning to improve inspection accuracy, reduce costs, and enhance efficiency. With further development and larger datasets, this approach could contribute significantly to the future of automated quality control in construction projects.

REFERENCES

Journal article

- Xu, Y. et al. (2021) 'Machine learning in construction: From shallow to deep learning', Developments in the Built Environment, 6, p. 100045. Available at:
- https://doi.org/10.1016/j.dibe.2021.100045.
- Baduge, S.K. et al. (2022) 'Artificial intelligence and smart vision for building and construction 4.0: Machine and deep learning methods and applications', Automation in Construction, 141, p. 104440. Available at: https://doi.org/10.1016/j.autcon.2022.104440.
- Ahmad, W., Ahmad, A., Ostrowski, K.A., Aslam, F., Joyklad, P. and Zajdel, P., 2021. Application of advanced machine learning approaches to predict the compressive strength of concrete containing supplementary cementitious materials. Materials, 14(19), p.5762. Available at: https://doi.org/10.3390/ma14195762
- Wei, W. et al. (2022) 'Integrated vision-based automated progress monitoring of indoor constructionusing mask region-based convolutional neural networks and BIM', Automation in Construction, 140, p. 104327. Available at: https://doi.org/10.1016/j.autcon.2022.104327.
- Luo, X. et al. (2018) 'Towards efficient and objective work sampling: Recognizing workers' activities insite surveillance videos with two-stream convolutional networks', Automation in Construction, 94, pp. 360–370. Available at: https://doi.org/10.1016/j.autcon.2018.07.011
- Li, J. et al. (2021) 'Evaluating the Work Productivity of Assembling Reinforcement through the ObjectsDetected by Deep Learning', Sensors, 21(16), p. 5598. Available at: https://doi.org/10.3390/s21165598.
- Chou, J.-S. and Nguyen, N.-M. (2022) 'Scour depth prediction at bridge piers using metaheuristicsoptimized stacking system', Automation in Construction, 140, p. 104297. Available at: https://doi.org/10.1016/j.autcon.2022.104297.
- Sri Kalyan, T., Zadeh, P.A., Staub-French, S. and Froese, T.M., 2016. Construction quality assessment using 3D as-built models generated with Project Tango. Procedia Engineering, 145, pp.1416-1423. Available at: https://doi.org/10.1016/j.proeng.2016.04.178
- Serradilla, O., Zugasti, E., Rodriguez, J. and Zurutuza, U., 2022. Deep learning models for predictive maintenance: A survey, comparison, challenges and prospects. Applied Intelligence, 52(10), pp.10934-10964. Available at: https://doi.org/10.1007/s10489-021-03004-y
- Kayhani, N., McCabe, B. and Sankaran, B., 2023. Semantic-aware quality assessment of building elements using graph neural networks. Automation in Construction, 155, 105054. Available at: https://doi.org/10.1016/j.autcon.2023.105054
- Katsamenis, I., Bakalos, N., Protopapadakis, E., Karolou, E.E., Kopsiaftis, G. and Voulodimos, A., 2023. Real time road defect monitoring from UAV visual data sources. In: Proceedings of the 16th International Conference on PErvasive Technologies Related to Assistive Environments, pp.603-609. https://doi.org/10.1145/3594806.3596561

- De Filippo, M., Asadiabadi, S., Kuang, J.S., Mishra, D.K. and Sun, H., 2023. AI-powered inspections of facades in reinforced concrete buildings. HKIE Transactions, 30(1), pp.1-14. https://doi.org/10.33430/V30N1THIE-2020-0023
- Ma, H. and Lee, S., 2022. Smart system to detect painting defects in shipyards: Vision AI and a deep learning approach. Applied Sciences, 12(5), p.2412. https://doi.org/10.3390/app12052412
- Trzeciakiewicz, M. et al. (2025) 'Automatic Drywall Analysis for Progress Tracking and Quality Controlin Construction', in Proceedings of the 20th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, pp. 707–716. Available at: https://doi.org/10.5220/0013302500003912.
- Raoofi, H. and Motamedi, A., 2020. Mask R-CNN deep learning-based approach to detect construction machinery on jobsites. In: 37th International Symposium on Automation and Robotics in Construction. Kitakyushu, Japan. https://doi.org/10.22260/ISARC2020/0154
- Vohra, M., Shehadeh, M., Abboud, M., Wang, Y. and Asmar, M., 2021. End-to-end real-time visual perception framework for construction automation. arXiv. https://doi.org/10.48550/arXiv.2107.12701
- Braun, A., Borrmann, A., Stilla, U. and Schrotter, G., 2020. Improving progress monitoring by fusing point clouds, semantic data and computer vision. Automation in Construction, 116, p.103210. https://doi.org/10.1016/j.autcon.2020.103210
- Ekanayake, B., Perera, S., Karunasena, G., Zhang, C., Alashwal, A. and Ekanayake, E., 2024. Deep learning-based computer vision in project management: Automating indoor construction progress monitoring. Project Leadership and Society, 5, p.100149. https://doi.org/10.1016/j.plas.2024.100149
- Braun, A. and Borrmann, A., 2019. Combining inverse photogrammetry and BIM for automated labeling of construction site images for machine learning. Automation in Construction, 106, p.102879. https://doi.org/10.1016/j.autcon.2019.102879
- Tang, X., Zhou, Q., Yu, M. and Wu, J., 2022. Benefits of terrestrial laser scanning for construction QA/QC: A time and cost analysis. *Journal of Management in Engineering*, 38(1). Available at: https://doi.org/10.1061/(ASCE)ME.1943-5479.0001012.
- Blinn, N., Issa, R.R.A., Hinze, J. and Olbina, S., 2017. Comparison of traditional laser scanning and mobile LIDAR technology for AECO applications. In: *Computing in Civil Engineering 2017*, American Society of Civil Engineers (ASCE), Gainesville, FL, USA, pp. 113–121. Available at: https://doi.org/10.1061/9780784480830.015.
- Faybishenko, B., Birkholzer, J., Pan, L. and Peterson, P., 2022. Challenging problems of quality assurance and quality control (QA/QC) of meteorological time series data. Stochastic Environmental Research and Risk Assessment, 36(4), pp.1049–1062. Available at: https://doi.org/10.1007/s00477-021-02106-w.
- Hoegh, K., Khazanovich, L. and Yu, H.T., 2011. Ultrasonic tomography for evaluation of concrete pavements. Transportation Research Record, (2232), pp.85–94. Available at: https://doi.org/10.3141/2232-09.

- Oliveira, R., Ferreira, M., Costa, A. and Mendes, R., 2023. Laser scanner in construction quality control of steel industrial buildings. Practice Periodical on Structural Design and Construction, 28(2). Available at: https://doi.org/10.1061/PPSCFX.SCENG-1389.
- Kwan, A. and Wong, H., 2005. Durability of reinforced concrete structures: theory vs practice. Proceedings of the Hong Kong Government Standing Committee on Concrete Technology Annual Concrete Seminar, pp.1–20. Available at: http://hdl.handle.net/10722/110796.
- Mavromatidis, L., Dauvergne, J., Lunazzi, R. and Batsale, J., 2014. First experiments for the diagnosis and thermophysical sampling using pulsed IR thermography from unmanned aerial vehicle (UAV). Quantitative InfraRed Thermography Journal. Available at: https://doi.org/10.21611/qirt.2014.213.
- Sreekanth, K. and Geetha, B., 2017. Introduction to artificial neural network (ANN) methods. International Journal of Advanced Research in Computer and Communication Engineering, 6(10), pp.342–345.
- Li, Y., Chen, C.L.P. and Zhang, T. (2022) 'A Survey on Siamese Network: Methodologies, Applications, and Opportunities', IEEE Transactions on Artificial Intelligence, 3(6), pp. 994–1014. Available at: https://doi.org/10.1109/TAI.2022.3207112.
- Wu, J. et al. (2018) 'Spot the Difference by Object Detection'. arXiv. Available at: https://doi.org/10.48550/arXiv.1801.01051.

Book

O'Brien, J.J., 1989. Construction Inspection Handbook. New York: Springer Science+Business Media

Conference proceedings

- Toyin, J.O. and Sattineni, A., 2025. Innovative approaches to QA/QC in construction: Current state. In: Proceedings of the 23rd CIB World Building Congress, 19–23 May 2025, Purdue University, West Lafayette, USA. CIB Conferences. Available at: https://docs.lib.purdue.edu/cib-conferences
- Melekhov, I., Kannala, J. and Rahtu, E., 2016. Siamese network features for image matching. In: Proceedings of the 23rd International Conference on Pattern Recognition (ICPR). Cancún, Mexico, 4–8 December 2016. IEEE, pp. 378–383. Available at: https://doi.org/10.1109/ICPR.2016.7899650
- Zhang, Y. et al. (2018a) 'Structured Siamese Network for Real-Time Visual Tracking', in V. Ferrari et al.(eds) Computer Vision ECCV 2018. Cham: Springer International Publishing (Lecture Notes in Computer Science), pp. 355–370. Available at: https://doi.org/10.1007/978-3-030-01240-3 22.
- Hoffer, E. and Ailon, N., 2015. Deep metric learning using triplet network. In: *Proceedings of the International Conference on Learning Representations (ICLR) 2015, Workshop Track.* San Diego, CA, USA, 7–9 May 2015. Available at: https://arxiv.org/abs/1412.6622
- He, A., Luo, C., Tian, X. and Zeng, W., 2018. A twofold Siamese network for real-time object tracking. Accepted by CVPR'18. Computer Vision and Pattern Recognition (cs.CV). Available at: https://doi.org/10.48550/arXiv.1802.08817

Schroff, F., Kalenichenko, D. and Philbin, J. (2015) 'FaceNet: A Unified Embedding for Face Recognition and Clustering', in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 815–823. Available at: https://doi.org/10.1109/CVPR.2015.7298682.

Website

www.openspace.ai

APPENDICES

APPENDIX 1: CODE

```
1 Setup
import matplotlib.pyplot as plt
import numpy as np import os
import random
import tensorflow as tf
from pathlib import Path from keras import applications
from keras import layers
from keras import losses
from keras import ops
from keras import optimizers
from keras import metrics from keras import Model
from keras.applications import resnet
target_shape = (200, 200)
2 Load Data set
!pip install gdown
Show hidden output
from google.colab import drive
drive.mount('/content/drive')
Expr Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
from pathlib import Path
cache dir = Path("/content/drive/MyDrive/250709 Siamese model trial")
anchor_images_path = cache_dir / "site"
positive_images_path = cache_dir / "bimmodel"
!gdown --id 16P159jXl_3Ch2rjsWIPeI_CgI33dtTLQ
!gdown --id 1NeZhqi1LJeFYrjE0zx8P-mqBJnTepSM3
lunzip -oq {cache_dir / "site.zip"} -d {cache_dir}
lunzip -oq {cache_dir / "bimmodel.zip"} -d {cache_dir}
def preprocess_image(filename):
    Load the specified file as a JPEG image, preprocess it and
    resize it to the target shape.
    image_string = tf.io.read_file(filename)
    image = tf.image.decode_jpeg(image_string, channels=3)
    image = tf.image.convert_image_dtype(image, tf.float32)
    image = tf.image.resize(image, target_shape)
return image
{\tt def\ preprocess\_triplets(anchor,\ positive,\ negative):}
    preprocess them.
    return (
        preprocess image(anchor),
```

```
preprocess_image(positive),
         preprocess_image(negative),
anchor images = sorted(
     [str(anchor_images_path / f) for f in os.listdir(anchor_images_path)]
positive_images = sorted(
    [str(positive_images_path / f) for f in os.listdir(positive_images_path)]
image_count = len(anchor_images)
anchor_dataset = tf.data.Dataset.from_tensor_slices(anchor_images)
positive_dataset = tf.data.Dataset.from_tensor_slices(positive_images)
rng = np.random.RandomState(seed=42)
rng.shuffle(anchor_images)
rng.shuffle(positive_images)
negative_images = anchor_images + positive_images
np.random.RandomState(seed=32).shuffle(negative_images)
negative_dataset = tf.data.Dataset.from_tensor_slices(negative_images)
negative_dataset = negative_dataset.shuffle(buffer_size=4096)
dataset = tf.data.Dataset.zip((anchor_dataset, positive_dataset, negative_dataset))
dataset = dataset.shuffle(buffer_size=1024)
dataset = dataset.map(preprocess_triplets)
train_dataset = dataset.take(round(image_count * 0.8))
val_dataset = dataset.skip(round(image_count * 0.8))
train_dataset = train_dataset.batch(32, drop_remainder=False)
train_dataset = train_dataset.prefetch(tf.data.AUTOTUNE)
val_dataset = val_dataset.batch(32, drop_remainder=False)
val_dataset = val_dataset.prefetch(tf.data.AUTOTUNE)
def visualize(anchor, positive, negative):
    """Visualize a few triplets from the supplied batches."""
     def show(ax, image):
         ax.imshow(image)
          ax.get_xaxis().set_visible(False)
          ax.get_yaxis().set_visible(False)
     fig = plt.figure(figsize=(9, 9))
     axs = fig.subplots(3, 3)
     show(axs[i, 0], anchor[i])
show(axs[i, 1], positive[i])
show(axs[i, 2], negative[i])
visualize(*list(train_dataset.take(1).as_numpy_iterator())[0])
Show hidden output
base_cnn = resnet.ResNet50(
     weights="imagenet", input_shape=target_shape + (3,), include_top=False
flatten = layers.Flatten()(base_cnn.output)
dense1 = layers.Dense(512, activation="relu")(flatten)
dense1 = layers.BatchNormalization()(dense1)
dense2 = layers.Dense(256, activation="relu")(dense1)
dense2 = layers.BatchNormalization()(dense2)
output = layers.Dense(256)(dense2)
```

```
embedding = Model(base_cnn.input, output, name="Embedding")
trainable = False
for layer in base_cnn.layers:
    if layer.name == "conv5_block1_out":
    trainable = True
     layer.trainable = trainable
class DistanceLayer(layers.Layer):
    This layer is responsible for computing the distance between the anchor embedding and the positive embedding, and the anchor embedding and the
     negative embedding.
    def __init__(self, **kwargs):
    super().__init__(**kwargs)
    def call(self, anchor, positive, negative):
    ap_distance = ops.sum(tf.square(anchor - positive), -1)
    an_distance = ops.sum(tf.square(anchor - negative), -1)
          return (ap_distance, an_distance)
anchor_input = layers.Input(name="anchor", shape=target_shape + (3,))
positive_input = layers.Input(name="nositive", shape=target_shape + (3,))
negative_input = layers.Input(name="nositive", shape=target_shape + (3,))
distances = DistanceLayer()(
     embedding(resnet.preprocess_input(anchor_input)),
embedding(resnet.preprocess_input(positive_input)),
     embedding(resnet.preprocess_input(negative_input)),
siamese_network = Model(
    inputs=[anchor_input, positive_input, negative_input], outputs=distances
class SiameseModel(Model):
     """The Siamese Network model with a custom training and testing loops.
     Computes the triplet loss using the three embeddings produced by the
    Siamese Network.
    The triplet loss is defined as: 
 _L(A, P, N) = max(\|f(A) - f(P)\|^2 - \|f(A) - f(N)\|^2 + margin, \theta)    """
     def __init__(self, siamese_network, margin=0.5):
          super().__init__()
self.siamese_network = siamese_network
          self.margin = margin
          self.loss tracker = metrics.Mean(name="loss")
    def call(self, inputs):
          return self.siamese_network(inputs)
    def train_step(self, data):
          with tf.GradientTape() as tape:
              loss = self._compute_loss(data)
          gradients = tape.gradient(loss, self.siamese_network.trainable_weights)
          self.optimizer.apply_gradients(
              zip(gradients, self.siamese_network.trainable_weights)
         self.loss tracker.update state(loss)
          return {"loss": self.loss_tracker.result()}
     def test_step(self, data):
         loss = self._compute_loss(data)
         self.loss tracker.update state(loss)
```

```
return {"loss": self.loss_tracker.result()}
     def _compute_loss(self, data):
          ap_distance, an_distance = self.siamese_network(data)
          loss = ap_distance - an_distance
         loss = tf.maximum(loss + self.margin, 0.0)
return loss
     @property
     def metrics(self):
          return [self.loss_tracker]
siamese_model = SiameseModel(siamese_network)
siamese_model.compile(optimizer=optimizers.Adam(0.0001))
siamese_model.fit(train_dataset, epochs=10, validation_data=val_dataset)
Show hidden output
sample = next(iter(train_dataset))
visualize(*sample)
anchor, positive, negative = sample
anchor_embedding, positive_embedding, negative_embedding = (
     embedding(resnet.preprocess_input(anchor)),
embedding(resnet.preprocess_input(positive)),
     embedding(resnet.preprocess_input(negative)),
Show hidden output
def compute_triplet_accuracy(dataset):
    correct = 0
total = 0
     for anchor, positive, negative in dataset:
    ap_distance, an_distance = siamese_model.siamese_network(
              [anchor, positive, negative]
          correct += tf.reduce_sum(tf.cast(ap_distance < an_distance, tf.int32)).numpy()</pre>
          total += anchor.shape[0]
     accuracy = correct / total
     return accuracy
train_acc = compute_triplet_accuracy(train_dataset)
val_acc = compute_triplet_accuracy(val_dataset)
print("Train accuracy:", train_acc)
print("Validation accuracy:", val_acc)
Show hidden output
```