BACHELOR’S THESIS IN COMPUTER SCIENCE AND INDUSTRIAL ECONOMICS

UNDERGRADUATE LEVEL 15 CREDITS

A Comparative Evaluation of Open-Source Digital Asset Management Systems

Exploring Organizational and Marketing Criteria for Process and Marketing Innovation in SMEs

ELLA KARLSSON

School of Industrial Engineering and Management
Royal Institute of Technology (KTH)

Abstract

(?)
• What is the topic area? (optional) Introduces the subject area for the project. • Short problem statement • Why was this problem worth a Master's thesis project? (i.e., why is the problem both significant and of a suitable degree of difficulty for a Master's thesis project? Why has no one else solved it yet?) • How did you solve the problem? What was your method/insight? • Results/Conclusions/Consequences/Impact: What are your key results/conclusions? What will others do based upon your results? What can be done now that you have finished - that could not be done before your thesis project was completed?

Keywords:
Digital Asset Management (DAM), Version Control, Metadata Management, Access Control, SMEs, Workflow Optimization

Sammanfattning

Nyckelord:

Acknowledgments

I would like to thank xxxx for having yyyy.

Contents

List of Figures

List of Tables

List of Acronyms and Abbreviations

AI	Artificial Intelligence
DAM	Digital Asset Management
DSR	Design Science Research
DT	Digital Transformation
ERP	Enterprise Resource Planning
IT	Information Technology
ML	Machine Learning
MCS	Management Control Systems
MDM	Metadata Management
RBAC	Role-based access control
RBV	Resource-Based View
SME	Small and Medium-sized Enterprises
UX	User Experience
VRIN	Valuable, Rare, Inimitable, Non-substitutable
YOLO	You Only Look Once

1 Introduction

To be added state-of-the-art one-stage object detection algorithm renowned for its efficiency and simplicity

1.1 Background

Digital Asset Management (DAM) emerged in the late 1990s as organizations began grappling with the rapid increase in digital content (Krogh, 2009). Early DAM systems were primarily on-premises solutions designed to store and manage assets such as images, videos, and documents. In the early 2000s, these systems transitioned to cloud-based platforms, offering improved scalability and accessibility (McCain et al., 2021).

More recently, the integration of Artificial Intelligence (AI) and machine learning (ML) has transformed DAM by automating key processes like image tagging, sorting, and categorization. Advanced computer vision techniques now enable systems to analyze and tag images automatically, reducing manual effort and increasing accuracy (Wu et al., 2022).

1.2 Problem

As bespoke manufacturers scale, managing digital assets—spanning product imagery, design renderings, and technical specifications—becomes essential for brand consistency and operational efficiency. However, most DAM solutions, especially open-source systems, lack the necessary automation, posing adoption and maintenance challenges for small and medium-sized enterprises (SMEs) with limited IT infrastructure. Wu et al. studied automated metadata annotation for cultural heritage and found that AI-generated captions often oversimplify context, such as describing a medieval knight merely as a “man on a horse” (Wu et al., 2022) This reflects similar challenges in design-driven manufacturing, where internal product terminology and industry-specific references require more precise and context-aware interpretation.

A core function of DAM is image tagging, sorting, and categorization, directly influencing asset retrievability and structural organization. Although AI has been integrated into some DAM solutions, these implementations typically rely on large pre-trained models that offer broad object classification rather than domain-specific tagging and vocabulary. Recent advancements in computer vision, particularly through algorithms such as YOLO (You Only Look Once), offer an opportunity to overcome these limitations. However, deploying a YOLO-powered system in this domain requires adapting the model to the specific features and vocabulary of the manufacturing sector. Rather than training a model from scratch—a process that demands extensive annotated data and computational resources—a more feasible approach is to fine-tune a pre-trained model using company-specific data.

1.3 Purpose

This study focuses on the metadata generation stage of Digital Asset Management (DAM), particularly automated image tagging using deep learning models.

The primary aim of this thesis is to assess the feasibility and impact of a YOLO-powered DAM system that has been fine-tuned on company-specific data to address the unique needs of premium manufacturing SMEs. The research will benchmark the performance of this fine-tuned system against a conventional open-source DAM platform (ResourceSpace), focusing on improvements in asset categorization accuracy and retrieval efficiency.

1.3.1 Technical Research questions

(a) To what extent does fine-tuning YOLOv11 and Faster R-CNN on company-specific manufacturing data improve object detection accuracy compared to a baseline model, in terms of precision, recall, and inference speed?

(b) What are the trade-offs between YOLOv11 and Faster R-CNN in terms of tagging quality, computational cost, and integration complexity within a DAM workflow?

(c) How do differences in model performance impact the usefulness of metadata for downstream tasks such as asset retrieval and categorization?

1.3.2 Business Research questions

Technological advancements alone do not guarantee successful integration. To complement this, the business perspective assesses the organizational and strategic impact after selecting the preferred DAM system. Specifically:

(d) What organizational and process changes are required to integrate AI-based image tagging into a manufacturing SME, and how do these changes affect knowledge structuring and internal workflows?

(e) What barriers emerge during implementation, and how are they influenced by the organization’s flexibility, strategic priorities, and project-based work culture?

(f) How does improved metadata generation contribute to long-term business value, such as brand consistency, operational scalability, and process innovation?

1.3.3 Societal Impact

Digital transformation has a significant impact on SMEs. These companies account for approximately 60% of total turnover and value-added contributions in Sweden's private sector, employing around 65% of the workforce (Tillväxtverket, 2021). The adoption of DAM systems is an integral part of this transformation, improving operational efficiency and reducing manual work, which contributes to broader economic growth. A cost-benefit analysis of 319 SMEs found that digital transformation enhances organizational resilience, reduces operational costs, and improves long-term scalability (Teng et al., 2022).

The stakeholders of this project?

This study is structured around a systematic process encompassing data collection, annotation, model fine-tuning, and testing. These phases represent essential steps that an SME would need to undertake if they were to implement a similar AI-based solution. By addressing both the positive impacts and the possible challenges, the aim is to to show if the benefits of adopting this solution justify the necessary investments and efforts. The project's outcomes are expected to contribute to academic knowledge in the field of AI-powered asset management, fostering further innovation.

1.3.4 Ethical considerations

Ethically, the project will investigate issues related to data privacy, transparency, and bias, which are critical in ensuring that automated systems operate fairly and without unintended consequences. These concerns are highlighted in the literature on AI ethics, which emphasizes the need for clear guidelines to mitigate risks associated with autonomous decision-making(Jobin et al., 2019).

1.3.5 Sustainability, and social considerations

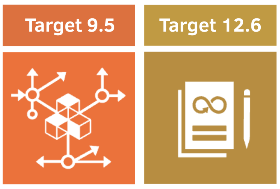

Figure 1-1: Sustainable Development Target 9.5 and 12.6

From a sustainability perspective, this research contributes to the United Nations Sustainable Development Goals (SDGs), specifically SDG 9, Industry, Innovation, and Infrastructure, and SDG 12, Responsible Consumption and Production, (United Nations, 2015). In relation to SDG 9, and more precisely target 9.5 as seen in Figure 1-1, the project seeks to enhance scientific research and upgrade the technological capabilities within industrial sectors. Similarly, under SDG 12 target 12.6 also shown in 1-1, this project supports sustainable business practices by optimizing digital asset management. By enhancing asset categorization and retrieval, the system makes it easier for companies to track and store metrics. This dual focus ensures that the technological advancements proposed are not only efficient and innovative but also ethically sound and socially beneficial.

Further reflection will be revisited in Section 6.4.

1.4 Goals

The primary goal is evaluating the feasibility of a YOLO-powered DAM system that has been fine-tuned using company-specific data, in comparison to the open-source solution ResourceSpace. To achieve this, the project has been divided into the following three sub-goals:

1. **Dataset Development and Annotation:** Develop a robust methodology for collecting a domain-specific dataset that accurately captures the visual and functional nuances of digital assets in premium manufacturing. The annotation process will involve:

   - Using bounding boxes to precisely delineate asset regions.
   - Assigning appropriate class labels using a standardized labeling schema to ensure consistency and relevance to the manufacturing domain.

   This dataset will serve as the foundation for model fine-tuning.

2. **Model Fine-Tuning and Optimization:** Fine-tune a pre-trained YOLO model on the annotated dataset. The objective is to enhance the model's accuracy in tagging, sorting, and categorizing.

   - Adjusting hyperparameters and leveraging transfer learning techniques.
   - Implementing regularization and validation strategies.

3. **Performance Benchmarking and Comparative Analysis:** Benchmark the performance of the fine-tuned YOLO-based DAM system against a conventional open-source DAM called ResourceSpace. Evaluation metrics will include:

   - Asset categorization accuracy.
   - Retrieval efficiency.
   - Overall system usability.

A comparative analysis will be conducted to assess whether the customized system offers significant improvements over traditional solutions. Resulting in practical recommendations and guidelines for manufacturing SMEs considering the adoption of AI-powered DAM.

1.5 Research Methodology

This research employs a mixed-methods approach to address both the technical performance of the system and stakeholder perspectives. Mixed-methods research combines quantitative techniques (e.g., controlled experiments and statistical analyses) with qualitative techniques (e.g., semi-structured interviews and thematic analysis) to provide a comprehensive evaluation of complex systems (Johnson and Onwuegbuzie, 2004).

Alternative methodologies—such as exclusively quantitative performance evaluations or purely qualitative case studies—were considered but ultimately rejected because they would not fully capture the multifaceted challenges of deploying an AI-powered system in a dynamic industrial environment.

1.5.1 Design Science Approach

Grounded in a pragmatic philosophy that emphasizes practical impact and utility, this study adopts the design science research (DSR) paradigm. DSR is particularly well-suited for technology-driven projects because it promotes the iterative design, development, and rigorous evaluation of IT artifacts to solve real-world problems (Hevner et al., 2004). In this project, the YOLO-powered DAM system represents the artifact developed and refined through iteration.

1.5.2 Quantitative and Qualitative Methods

Controlled experiments will be conducted to measure key performance metrics—such as asset categorization accuracy, retrieval efficiency, and overall system usability. Statistical analysis w ill be used to validate the improvements brought about by model fine-tuning, following best practices in empirical research (Creswell, 2014; Yin, 2014). Complementing this, qualitative methods will capture contextual insights and stakeholder perspectives. Semi-structured interviews and thematic analysis will be employed to understand user experiences and organizational challenges associated with implementing the DAM system. Moreover, to develop a standardized labeling schema for the dataset, a targeted collaboration with a designated expert from the company will be undertaken. This focused approach is preferred over a large-scale survey. Not all employees interact with digital assets and the expert can ensure domain-specific terminology is accurately captured and applied consistently during annotation.

1.6 Delimitations

This thesis focuses exclusively on evaluating a YOLO-powered digital asset management system for premium manufacturing SMEs. The study is limited to a specific company's environment and a predefined dataset.

The research investigates only the fine-tuning of an existing pre-trained YOLOv11 model. Training a model from scratch, which requires vast amounts of data and computational resources, is beyond the scope of this project. Instead of conducting a large-scale survey, the study uses semi-structured interviews with key stakeholders—particularly a designated domain expert—to develop a standardized labeling schema.

This focused approach is chosen because only a few employees directly manage digital assets. The assessment will concentrate on technical performance indicators such as asset categorization accuracy, retrieval efficiency, and overall system usability. Broader issues such as integration with other enterprise systems and macroeconomic impacts are beyond the scope of this project.

1.7 Structure of the thesis

This thesis is organized into the following main chapters, excluding the introductory chapter, references, and appendices; Chapter 2 provides the necessary background and reviews related work, establishing the context for DAM and identifying the key gaps this project addresses. Chapter 3 outlines the methodology—including the design science approach, mixed-methods strategy, data collection, experimental design, and evaluation criteria—used to assess the system. Chapter 4 details the implementation, covering system design, model fine-tuning, dataset development, and the technical setup for testing. Chapter 5 presents the results and analysis, discussing both quantitative metrics and qualitative insights to evaluate whether the project's goals have been met. Finally, Chapter 6 summarizes the key findings, reflects on the limitations of the study, and outlines potential directions for future work.

2 Background

2.1 Artificial Inteligence

Artificial Intelligence (AI) is a field of computer science that focuses on systems built on algorithms, which are formalized sets of instructions that process input data to produce outputs (Khanam et al., 2024a). Machine Learning (ML), a subset of AI, represents a shift away from manually encoded rules toward data-driven learning. Instead of being explicitly programmed for specific tasks, ML models identify patterns in large datasets and use statistical techniques to make predictions or classify new data.

Khanam et al. (2024a) describe deep learning (DL) as a machine learning approach that utilizes multi-layered computational models to extract patterns from data at varying levels of abstraction. Inspired by the human brain, DL models excel at recognizing intricate patterns in large datasets. (Soori et al., 2023) further eplains that within DL, different neural network architectures are designed to process specific types of data and perform specialized tasks. One of the most effective architectures for structured, grid-like data—such as images—is the Convolutional Neural Network (CNN). CNNs employ convolutional operations to automatically learn spatial hierarchies of features, allowing them to capture patterns and structures in data with high accuracy. As a result, CNNs have become a cornerstone of computer vision, powering applications in object detection, image classification, and other visual recognition tasks (Goodfellow et al., 2016, pp. 326-328).

2.1.1 Object Detection

Object detection involves both the ability to recognize the classes of multiple objects in an image and determining their positions, whereas image classification assigns a single class to the entire image without distinguishing individual objects.

Zhang et al. (2025) outline how DL-based object detection methods are primarily divided into two categories: two-stage and single-stage networks. Two-stage networks, such as Region-Based Convolutional Neural Networks (R-CNNs), rely on generating region proposals before classifying and refining object locations. In contrast, single-stage networks, such as You Only Look Once (YOLO), eliminate this intermediate step by predicting object classes and bounding boxes in a single pass. This approach significantly improves detection speed and efficiency. As Zhang et al. (2025) emphasize, single-stage models have become widely adopted in various industries due to their ability to perform real-time object detection accurately.

2.1.2 Anchor-free detection models

A bounding box defines an object's position and size within an image using four coordinates. In object detection, it is paired with a class label and a confidence score, indicating both the object's category and the model's certainty in its prediction. These boxes act as ground-truth references in training data, helping models learn to localize objects accurately. (Li et al., 2022). The prediction represents the final output of an object detection model as illustrated in Figure 2-1.

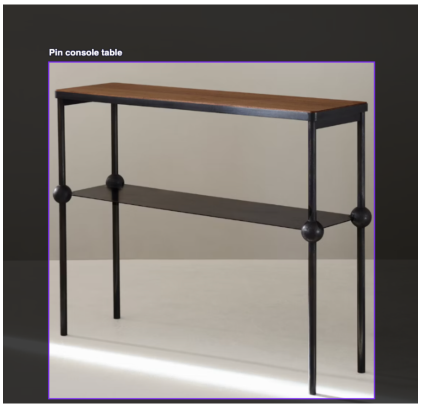

Figure 2-1: Bounding box for table with legs.

Vina (2024) describes the shift from anchor-based to anchor-free object detection as a major advancement in the field. Traditional anchor-based detectors, such as YOLOv4 and its predecessors in Table 2.1, rely on predefined anchor boxes—fixed-size reference shapes placed across an image at different aspect ratios—to estimate object locations. The model does not predict bounding boxes directly but instead modifies the closest anchor to better fit detected objects. Anchor-free models simplify detection and improve speed—critical for real-time tasks like autonomous driving and surveillance. Their keypoint-based approach enhances flexibility, making them better at detecting small, irregular, or occluded objects, especially in cluttered environments where anchor-based methods struggle (Wang et al., 2024b).

2.2 The Architecture of a Convolutional Neural Network

Prince (2023) highlights three key characteristics of digital images that necessitate the use of specialized model architectures. First, images are inherently high-dimensional. For instance, a standard 224×224 pixel image with three color channels (RGB) results in over 150,000 input values. Processing such a large number of inputs with fully connected neural networks would require an impractically high number of parameters. Second, there is a strong correlation between neighboring pixels, as local regions often form meaningful patterns and structures. Lastly, images tend to be robust to small spatial shifts—their content remains recognizable even when objects within them are slightly moved. For instance, if a chair appears slightly to the left or right in different images, we still recognize it as the same object. However, a fully connected model would need to learn how to identify the chair in every possible position from scratch. CNNs such as YOLO and Faster R-CNN avoid this problem by using filters that can detect patterns no matter where they appear in the image. This makes them far more parameter-efficient and better suited for visual tasks like object detection (Prince, 2023).

At a fundamental level, CNNs process input through sequential stages, using convolution to detect spacial features, pooling to reduce dimensionality, and activation functions to introduce non-linearity Khanam et al. (2024b). Spatial features can be textures, lines and color variations in the input. With effective training, the network learns to recognize these attributes regardless of their location within an image (Verdhan, 2021, Chapter 2).

2.2.1 The Convolutional Operation

CNNs extract features from images by applying an operation known as convolution (Prince, 2023, p. 170). Convolution involves sliding a learnable weight matrix, referred to as a kernel or filter, across the input. At each position, the kernel computes a weighted sum over a local neighborhood of the image, making it possible to detect spatial patterns. Figure 2-2 illustrates this concept. In practice, one often pads the input with zeros (padding) so that the kernel can be applied near image borders without reducing spatial dimensions. Another key hyperparameter is the stride, which specifies how far the kernel moves at each step (Prince, 2023, p. 165)

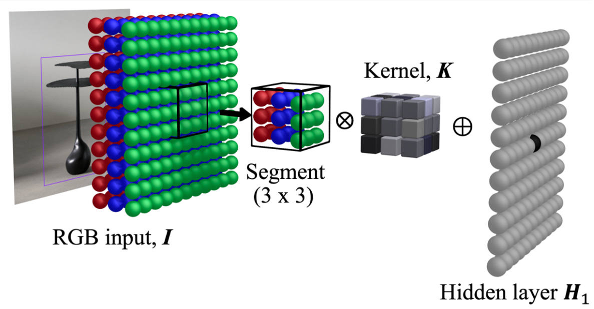

Figure 2-2: A simplified 2D convolution applied to an RGB image (adapted from (Prince, 2023)).

Let I be the input image, structured as three channels (red, green, and blue). Consider a $3 \times 3 \times 3$ kernel. At each spatial position, element-wise multiplication is performed between the kernel weights and a 3×3 segment from each of the three channels. The products are summed together and then combined with a bias term, producing a pre-activation value that is typically passed through a non-linear function such as ReLU. By shifting the kernel step by step over the height and width of the image, one obtains a two-dimensional feature map. To produce multiple output channels, different kernels run in parallel. Each filter generates its own 2D feature map, and stacking these maps forms a three-dimensional activation tensor, often written as H_1. Equation (1) demonstrates how the output at position (i, j) can be computed for an RGB input and a 3×3 kernel:

$$h_{ij} = a\Big(b + \sum_{c=1}^{3}\sum_{m=1}^{3}\sum_{n=1}^{3} I_{c,\,i+m-2,\,j+n-2} \cdot K_{c,\,m,\,n}\Big) \quad (1)$$

where $I_{c,\,i,j}$ denotes the pixel value from channel c at position (i, j), $K_{c,\,m,n}$ is the kernel weight for channel c at offset (m, n), b is a learnable bias term, and $a(\cdot)$ represents the chosen activation function (Prince, 2023, p. 170).

2.2.2 The Pooling Operation

Pooling is a downsampling operation in CNNs that reduces the spatial dimensions of feature maps while preserving essential features. This improves computational efficiency and makes the network less sensitive to small spatial shifts. The most common method, max pooling, slides a fixed-size window over the feature map and retains only the maximum value in each region as seen in Figure 2-3 (Prince, 2023, p. 163).

$$X_{4,4} = \begin{bmatrix} 9 & 4 & 1 & 5 \\ 1 & 1 & 1 & 1 \\ 1 & 2 & 1 & 4 \\ 1 & 3 & 3 & 7 \end{bmatrix} \rightarrow Y_{2,2} = \begin{bmatrix} 9 & 5 \\ 3 & 7 \end{bmatrix}$$

Figure 2-3: Max pooling applied to a 4×4 matrix X resulting in a 2×2 matrix Y.

The latest YOLO models developed by Jocher and Ultralytics (2025) extend this concept with the SPPF (Spatial Pyramid Pooling Fast) block, which increases the receptive field through repeated pooling. Figure 2-4 shows its placement in the Neck in the architecture. The operation is defined as:

$$\text{SPPF} = \text{Conv}_{1\times1}\left(\text{Concat}(X,\ P_1,\ P_2,\ P_3)\right) \quad (2)$$

where X is the input feature map, first passed through a 1×1 convolution to reduce channel dimensions. $P_1 = \text{MaxPool}_{5\times5}(X)$, $P_2 = \text{MaxPool}_{5\times5}(P_1)$, and $P_3 = \text{MaxPool}_{5\times5}(P_2)$. All outputs (X, P_1, P_2, P_3) are concatenated along the channel dimension and passed through a second 1×1 convolution. This design allows the model to capture multi-scale contextual information from increasingly larger regions while maintaining spatial resolution, which improves object detection performance, especially for small or partially occluded objects (Jocher and Ultralytics, 2025).

2.2.3 Activation Functions

The Ultralytics YOLO architecture by Jocher and Ultralytics (2025) primarily uses the Sigmoid Linear Unit (SiLU), also known as *Swish*, as its default activation function. It is defined as

$$\mathrm{SiLU}(x) = x \cdot \sigma(x) = \frac{x}{1 + e^{-x}}, \tag{3}$$

where $\sigma(x)$ represents the sigmoid function. SiLU in Equation (3) offers a smooth non-linearity that helps the model train more efficiently and maintain stronger gradient signals in deep layers.

A simpler alternative, ReLU (Rectified Linear Unit), in Equation (4),

$$\mathrm{ReLU}(x) = \max(0, x). \tag{4}$$

is used in certain parts of the network that benefit from faster computation and sparser activations. Additionally, some layers omit activations altogether to maintain strictly linear connections. This is sometimes useful in residual paths or when merging feature maps. However, SiLU remains the primary activation due to its observed advantages in training stability and overall performance (Jocher and Ultralytics, 2025).

2.2.4 Structural Components of the YOLO Architecture

The three-part YOLO structure consists of Backbone, Neck, and Head—as shown in Figure 2-4. The Backbone extracts features using convolutional layers and downsampling, generating hierarchical feature maps. The Neck refines these features through the SPPF block for multi-scale detection and the C2PSA module to enhance the recognition of occluded objects. Upsampling and feature concatenation further improve resolution and information retention. Finally, the Head produces the model's output, predicting class probabilities and bounding boxes across three detection layers (small, medium and large), each specialized for different object sizes (Hidayatullah et al., 2025).

The C3k2 module, used in both the Backbone and Neck (Figure 2-4), acts like a compact feature extractor. It splits the input in half: one part flows through unchanged, while the other is processed by a stack of C3k blocks—convolutions with varied kernel sizes to capture both fine and coarse spatial patterns. The two paths are merged and compressed through a 1×1 convolution (Hidayatullah et al., 2025).

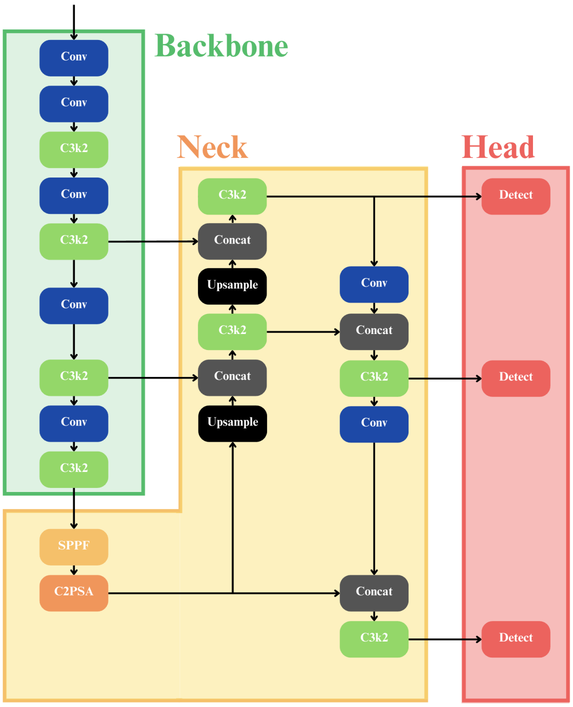

Figure 2-4: The architecture of YOLOv11, illustrating its three main components: Backbone, Neck, and Head (adapted from (Hidayatullah et al., 2025)).

The C2PSA (Cross-Stage Partial with Position-Sensitive Attention) module following after the SPPF block in Figure 2-4 extends the Cross-Stage Partial (CSP) design with a more expressive attention mechanism known as Position-Sensitive Attention (PSA). While C3k2 in captures features through varied convolution kernels, C2PSA uses attention to focus on relevant spatial patterns—especially useful for detecting large objects at low resolutions (Jocher and Ultralytics, 2025).

Upsampling increases the spatial resolution of feature maps to restore details lost during downsampling. YOLO typically employs nearest-neighbor upsampling, duplicating pixels to double feature map dimensions (Jocher and Ultralytics, 2025). The subsequent concatenation merges these upsampled feature maps with earlier layers, enriching feature representations and improving multi-scale object detection capability (Figure 2-4; Hidayatullah et al., 2025).

2.2.5 YOLOv11 model

The YOLOv11 model, developed by Ultralytics marks the latest milestone in the continuous evolution of the YOLO series, building on a decade of refinement and optimization, as summarized in Table 2.1. Since its introduction by Redmon et al. (2016), it has revolutionized real-time object detection with its single-stage pipeline, offering a faster and more efficient alternative to traditional region-based approaches like R-CNNs.

Release	Key capabilities
V1 JUN 2015	Darknet. A single-stage object detector with basic classification (Redmon et al., 2016).
V2 DEC 2016	Darknet. Object detection. Darknet-19 architecture, anchor boxes, and higher resolution inputs (Redmon and Farhadi, 2016).
V3 MAR 2018	Darknet. Object detection. Darknet-53 network & multi-scale predictions for varying object sizes. (Redmon and Farhadi, 2018).
V4 APR 2020	Darknet. Object detection. Basic object tracking with BCSPDarknet53 and SPP. (Bochkovskiy et al., 2020).
V5 JUN 2020	PyTorch. Object detection. Basic instance segmentation. Multi-GPU support, and exports (Ultralytics, 2020).
V6 SEP 2022	PyTorch. Object detection, instance segmentation, a reparameterizable backbone, anchor aided training (AAT). (Li et al., 2022).
V7 JUL 2022	PyTorch. Object detection, tracking & instance segmentation. (Wang et al., 2022).
V8 JAN 2023	PyTorch. Anchor-free object detection, instance & panoptic segmentation, NVIDIA GPUs, Jetson. (Ultralytics, 2023).
V9 FEB 2024	PyTorch. Anchor-free detection & instance segmentation. PGI for better gradient reliability. GELAN network (Wang et al., 2024b).
V10 MAY 2024	PyTorch. Anchor-free detection & NMS-free training (Wang et al., 2024a).
V11 SEP 2024	PyTorch. Anchor-free & oriented object detection (OBB), instance segmentation, pose estimation. (Ultralytics Inc., 2025).
V12 FEB 2025	PyTorch. Anchor-free detection, OBB, instance segmentation, Area Attention Mechanism, pose estimation, R-ELAN. (Ultralytics Inc., 2025).

Table 2.1: Summary of YOLO Model Evolution

Early versions of YOLO were built on the Darknet framework, developed by Joseph Redmon, with core implementations written in C and CUDA for fast GPU execution. A framework is a pre-built structure that simplifies software development by providing reusable code, tools, and libraries allowing developers to focus on higher-level abstraction. As shown in Table 2.1, the transition to PyTorch occurred with YOLOv5, developed by Ultralytics. PyTorch, originally introduced by Facebook AI Research (FAIR), offered a more flexible and scalable environment, facilitating development in Python and enhancing integration with mainstream deep learning research (Ultralytics, 2020).

Sapkota et al. (2025) conducted a comprehensive review of YOLO-based object detection applications, highlighting its extensive adoption across multiple domains, including healthcare (e.g., pill identification, diagnostics), surveillance (e.g., face mask detection, home security), autonomous vehicles, and industrial quality control. The study underscores YOLO's efficiency in real-time processing, making it a preferred choice for applications requiring rapid inference.

While YOLO excels in speed, its grid-based detection approach and anchor-free methodology maintained in YOLOv6 and subsequent models introduce inherent limitations. Both Sapkota et al. (2025) and He et al. (2024) note that, despite its computational efficiency, YOLO may struggle with fine-grained detail detection, making it less suitable for tasks requiring high-resolution texture analysis, such as road damage assessment or material surface inspection (Angulo et al., 2019). While this thesis primarily addresses the application of YOLO within bespoke manufacturing, insights into the limitations remain highly relevant, particularly in scenarios where accurate detection and classification of subtle material textures effect performance.

The trade-off between speed and accuracy is further emphasized in comparative analyses, such as Rane (2023), which contrasts YOLO with Faster R-CNN. While YOLO excels in inference speed—making it well-suited for real-time applications such as inventory management, checkout automation, and e-commerce visual search—Faster R-CNN offers superior object localization and classification accuracy. This aligns with the findings of Sapkota et al. (2025), making it the preferred choice for scenarios demanding precise differentiation and high recall, such as medical imaging. However, Faster R-CNN's reliance on a region proposal network (RPN) results in significantly higher computational demands, limiting its viability for real-time deployment (Rane, 2023).

In contrast, the study by Karbouj et al. (2024) on object detection for screw head identification in disassembly systems presents a different perspective. Their findings demonstrate that YOLOv5 outperforms Faster R-CNN across multiple key metrics, including precision, recall, inference speed (FPS), and training efficiency. This discrepancy arises from the nature of the application and dataset size. As previously discussed by Rane (2023) Faster R-CNN tends to perform better in tasks requiring high-detail object recognition. The RPN helps it generalize more effectively when training data is limited, making it particularly useful for small datasets with high precision requirements. Conversely, YOLO's ability to efficiently learn broad patterns makes it a superior choice for large-scale, high-variance datasets. The findings of Karbouj et al.

(2024) reinforce this perspective, demonstrating YOLOv5's balance between computational speed and adaptability, making it particularly effective in real-time, resource-constrained environments.

(Alif and Hussain, 2025)

As for relating to this thesis. there is limited research on the use of YOLO directly relating for Digital Asset Management (DAM) applications. with only one identified study—Angulo et al.

citeSapkota2025YOLOv11.

The improvements of Yolov11 OLOv11 outperformed previous versions in mean average precision (mAP), recall, and precision, demonstrating superior object detection performance. The recall rate, which measures how well the model detects all ground-truth objects, was highest for YOLOv11 (64.8YOLOv11 also exhibited fewer false detections compared to its predecessors. YOLOv11 displayed higher attention concentration on relevant objects, meaning it focused better on wires and transformers, reducing errors in object localization.

2.3 Object Detection with YOLOv11

construction of a object detection dataset

image preprocessing,

model training using the object detection training dataset,

and validation of results using a verification dataset

YOLO's backbone network has undergone substantial advancements, integrating deeper feature fusion and multiscale feature extraction to enhance its capability for power equipment object detection.

Starting from YOLOv8 [13], the series adopted an anchorfree mechanism for the first time, allowing greater adaptability to detect power equipment targets of varying sizes.

Since YOLOv5 [12], the algorithm has significantly improved detection efficiency and accuracy through the introduction of the CSPNet framework, which optimizes feature propagation and network capacity

updates to the YOLO series have included innovative enhancements to the loss function, further refining the model's detection precision. While the original YOLO algorithm offered remarkable detection speed, its accuracy lagged behind two-stage detection algorithms. H

The incorporation of a spatial pyramid pooling (SPP) layer into the backbone network further expanded the model's receptive field, enhancing its feature extraction capabilities. YOLOv5 advanced these capabilities by adopting the C3 module in its backbone network, which reduced computational complexity and improved inference speed. It also introduced Mosaic data augmentation, particularly Mosaic4, which combines and transforms four images randomly to enhance feature representation and model learning. Adaptive anchor box optimization was added, enabling the model to better handle objects of different sizes.YOLOv8 refined the architecture further by replacing the C3 module with the C2f module, enhancing feature extraction efficiency.

It also introduced an Anchor-Free detection mechanism to improve the detection of small targets. The Mosaic augmentation process was optimized to exclude its use in the final ten training epochs, thereby improving model generalization. Additionally, task-specific loss optimizations were integrated to further enhance detection performance. YYOLOv9 [16] introduced progressive gradient integration (PGI), addressing limitations of deep supervision in extremely deep architectures and making lightweight architectures more practical. A new network architecture, called generalized high-efficiency layer aggregation network (GELAN), was proposed. GELAN integrates cross stage partial network (CSPNet) and efficient layer aggregation network (ELAN) designs, balancing model lightweight design, inference speed, and accuracy. Crossstage partial connections were employed to link feature maps across stages, enriching semantic information and improving

Among these, You only look once (YOLO) , a real-time object detection algorithm, has gained widespread attention. Unlike traditional methods, YOLO eliminates the need for pre-generated candidate regions, directly predicting the class and location of targets within an image. Since its inception in 2015, YOLO has undergone significant advancements, with the latest version, YOLOv11, demonstrating substantial improvements in detection speed and performance

Architectures within the object detection domain can be classified into single-stage or two-stage detectors

YOLO significantly enhances the speed, efficiency, and accuracy of medical object detection compared to traditional methods.

Typical neural network: Input neuron (each connected to each next leyer)- hidden leyer - ouptut layers

In a concolutional network it is not mandatory all neurons are connected to each in the next hidden layer.

Filters: the fixed square called a patch or local receptive field

Feature map: The feature map is the output of one filter applied to the previous layer.

The filter moves across the input layer. (multyply the values within th filter with the values in the inpur layer). A new matrix with less diemnsions is compartmentalized

input layer is called local receptive fieilds.

Activation and Pooling layers: Activation: transforming to the output using Activation functional like Resulting(discard the negative values and replace them with zeros)

Pooling: The feature map dimensionallytyy is reduced using pooling (only improtant features remain... man, min pooling etc i.e to the only largest)

1. convolutional layers 2. ppoling layers 3. fully connected layers

2.4 LOSS

The YOLOv11 object detection method enhances its performance by minimizing a comprehensive loss function that integrates multiple components. This loss function encompasses distributed focal loss, bounding box regression loss, and class probability loss. The optimization process involves combining these individual loss components and employing advanced optimization algorithms to refine the model's performance in object detection tasks

2.5 Digital Asset Management

Krogh (2009) describes DAM as an essential framework for protecting, organizing, and prolonging the usability of digital files by emphasizing metadata, suitable file formats, and efficient workflows. As shown in Figure 2-5, five interconnected stages—creation, management, distribution, archiving, and retrieval—collectively ensure that digital assets remain discoverable and relevant long after their initial production.

Although Krogh does not explicitly align his approach with the Resource-Based View (RBV), his emphasis on preserving assets as integral organizational resources parallels RBV's tenet that competitive advantage relies on valuable, rare, inimitable, and non-substitutable (VRIN) capabilities (Barney, 1991). By structuring DAM processes around rigorous metadata management, secure storage, and ongoing accessibility, organizations can treat their digital repositories as strategic assets, safeguarding long-term benefits that are difficult for competitors to replicate.

2.5.1 Choosing a DAM and the key tasks

What tools are available in DAM? Bechmark? What are the most important shit in it? What do most companies need? What do they usually have and how or why do they choose to adopt a DAM

A missing perspective is

2.5.2 Technological Tools Demand Continuous Organizational Adaptation

Love and Matthews (2019) identify a critical gap in the construction industry: knowing "why" to adopt digital technologies is relatively straightforward, but knowing "how" to translate technological potential into real value remains largely underexplored. Their case studies underscore the fact that digital transformation does not happen automatically; organizations must actively invest in processes such as benefits management and the development of a Business Dependency Network (BDN) to realize tangible gains from their digital initiatives (Love and Matthews, 2019).

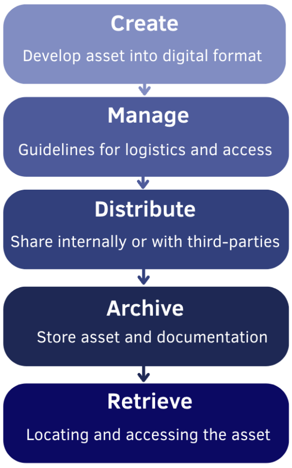
Create
Develop asset into digital format
Manage
Guidelines for logistics and access
Distribute
Share internally or with third-parties
Archive
Store asset and documentation
Retrieve
Locating and accessing the asset

Figure 2-5: Illustrating the five main stages of DAM.

In a broader context, Hanelt et al. (2020) posit that digital transformation (DT) goes beyond any single disruptive episode; it is a continual, structural adjustment propelled by digital technologies. Their systematic review of 279 peer-reviewed articles frames DT across three dimensions—Contextual Conditions (e.g., technological advances, shifting consumer habits), Mechanisms (e.g., the innovative strategies organizations adopt), and Outcomes (e.g., changes to organizational structures and industry norms). By proposing a typology that spans technology impact, compartmentalized adaptation, systemic shift, and holistic co-evolution, they challenge the idea of one-off change, advocating instead for an iterative, agile approach to transformation (Hanelt et al., 2020).

Taken together, these two perspectives highlight that while there is strong motivation to deploy new technologies ("why"), sustained, organization-wide benefits only materialize when there is a concerted effort to integrate, evaluate, and adapt these digital tools in an ongoing manner ("how"). Both studies imply that true success hinges on long-term structural and cultural shifts rather than static, one-off solutions.

that the promise of DAM is not unlocked simply by adopting new technology but only when companies embrace two fundamental principles. First, that technology alone does not create value but must be accompanied by organizational process reengineering, and second, that the benefits of DAM are maximized only through continuous strategic governance to monitor and sustain its impact

A missing perspective in

Nevertheless, some scholars argue that resource possession alone does not guarantee successful digital transformation. Civelek et al. (2023) found no significant link be- tween dynamic capabilities—a key aspect of RBV that involves adapting, integrating, and reconfiguring resources—and successful digital transformation among Czech manu- facturing SMEs. Their findings suggest that merely possessing dynamic capabilities is insufficient for digital transformation unless supported by complementary factors such as digital literacy and IT infrastructure matu- rity.

2.5.3 Why to make our own and not use a service

Bynder

Adobe Experince Manager

Cloudinary: custom pricing for enterprise solutions.

Adobe sensei enerally means auto-tagging images based on recognizable generic objects, scenes, and concepts. It typically uses generalized, pre-trained models that identify common objects'

most DAM platforms rely on third-party integrations for company-specific tagging

Clarifai Custom Models Provides APIs that integrate into DAM platforms.

Amazon Rekognition Custom Labels: Pay-per-use

Google Vertex AI (formerly AI Platform Vision) Pricing depends on training hours and predictions Custom vision API: Trained specifically on your images and product labels.

Microsoft Azure Custom Vision: Training: 20 dollaar per compute hour

Integrates via REST API to enhance tagging accuracy in DAM solutions.

CV consutling

Image annotation

Different types of CV:

?? is an image **??** is a table

2.5.4 Major background area#1#1

Recent studies have demonstrated the effectiveness of various AI techniques in image tagging. Zhang et al. (2019) showcased the application of convolutional neural networks (CNNs) for automatic image classification in DAM systems, achieving an accuracy of 92% on a diverse dataset of digital assets

This work was further extended by Li and Chen (2020), who integrated attention mechanisms into CNNs, improving the model's ability to focus on salient features and increasing tagging accuracy to 95%

The YOLO (You Only Look Once) algorithm has also been applied successfully in DAM contexts. Wang et al. (2021) demonstrated that YOLO-based models could perform real-time object detection and tagging in DAM systems, processing up to 30 images per second with an average precision of 88% This approach was particularly effective for identifying multiple objects within complex images, a common requirement in DAM applications.

Transformer-based models have recently gained traction in image tagging for DAM systems. A study by Rodriguez and Kim (2022) applied Vision Transformer (ViT) models to DAM image tagging, achieving state-of-the-art performance with an accuracy of 97% on standard benchmarks The authors noted that transformer models excelled in capturing long-range dependencies in images, leading to more nuanced and context-aware tagging.

While AI-powered image tagging offers significant benefits, it also presents several challenges. Data requirements pose a significant hurdle, as highlighted by Brown et al. (2020), who found that AI models required at least 10,000 labeled images per category for optimal performance in domain-specific DAM applications

Error rates and handling domain-specific content remain ongoing challenges. A comprehensive study by Thompson et al. (2021) analyzed error patterns in AI-powered image tagging across various industries, revealing that error rates increased significantly (up to 25%) when dealing with highly specialized or technical imagery

To address this issue, Nguyen and Patel (2022) proposed a hybrid approach combining pre-trained models with domain-specific fine-tuning, reducing error rates by 40% in niche industries such as medical imaging and aerospace engineerin

Despite these challenges, the benefits of AI-powered image tagging in DAM systems are substantial. A large-scale study by Garcia et al. (2023) across 500 organizations found that implementing AI-powered tagging led to a 60% reduction in manual tagging time and a 35% improvement in asset discoverability

Entangled states are an important part of quantum cryptography, but also relevant in other domains. This concept might be relevant for neutrinos, see for example [2].

Scheme

2.5.5 The YOLO model

As demonstrated in table 2.1 the YOLO series has evolved significantly since its inception, introducing progressive improvements in object detection, computational efficiency, and feature extraction. YOLOv11 is the best choice for the project due to its superior accuracy, efficiency, and versatility. As Khanam and Hussain (2024) highlight, its architectural upgrades enhance feature extraction while minimizing computational costs, making it ideal for real-time applications requiring both speed and precision (Khanam and Hussain, 2024).

Beyond object detection, YOLOv11 supports instance segmentation, pose estimation, and oriented object detection, offering greater adaptability to the project's needs. Its optimized balance of accuracy and processing speed ensures strong performance across different computing environments, from edge devices to high-performance systems, making it the most effective solution

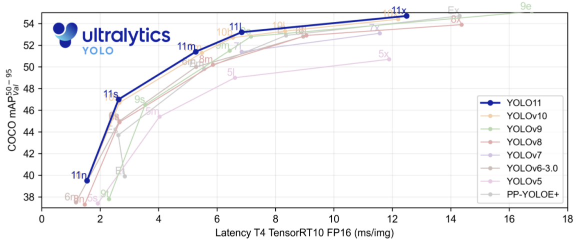

Figure 2-6: YOLOv11 performance comparison (Ultralytics Inc., 2025).

The selection of YOLOv11 for the project is driven by its superior architectural enhancements, versatile task support, and optimized balance between accuracy and efficiency. Each version has incorporated refinements aimed at enhancing real-time performance, with YOLOv11 representing the most advanced iteration to date (Khanam and Hussain, 2024).

2.6 Major background area#2

The application of AI-powered image tagging in DAM systems extends beyond large corporations to small and medium-sized enterprises (SMEs), particularly in premium manufacturing sectors. A case study by Hoffmann and Schulz (2022) examined the implementation of AI-powered DAM in a high-end carpentry company similar to Veermakers The study found that AI-assisted tagging improved product catalog management efficiency by 45% and reduced time-to-market for new designs by 30%.

However, Chen et al. (2023) noted that SMEs in specialized manufacturing often face unique challenges in adopting AI-powered DAM systems, including limited datasets and highly specific visual content. To address these issues, the authors proposed a transfer learning approach, adapting pre-trained models to domain-specific tasks with minimal additional data, achieving a 75% reduction in required training data while maintaining 90% of the original accuracy.

While academic research has made significant strides in advancing AI-powered image tagging techniques, commercial implementations often lag behind in adopting cutting-edge methods. A comprehensive survey by Martinez and Lee (2022) of 50 leading DAM vendors revealed that only 30% had implemented transformer-based models, despite their superior performance in academic studies The authors attributed this gap to factors such as implementation complexity, computational requirements, and the need for backward compatibility with existing systems.

2.6.1 Major background area#2#1

The integration of AI-powered image tagging in DAM systems raises important ethical, societal, and legal considerations. Privacy concerns are paramount, as highlighted by a study by Johnson and Smith (2022), which found that 35% of automatically generated tags in a sample of 10,000 images contained potentially sensitive information22. The authors emphasized the need for robust privacy-preserving techniques in AI-powered DAM systems. Algorithmic bias presents another significant challenge. Research by Park et al. (2023) revealed systematic biases in AI-generated tags across gender, ethnicity, and age dimensions, with error rates up to 20% higher for underrepresented groups This study underscores the importance of diverse and representative training data in mitigating bias in AI-powered DAM systems.

2.6.2 Major background area#2#2

The potential impact on employment is also a concern. While Garcia et al. (2023) found that AI-powered tagging led to significant efficiency gains, they also noted a 15% reduction in human tagging roles across surveyed organizations However, the same study observed a 10% increase in higher-skilled positions related to AI model management and quality assurance, suggesting a shift rather than a net loss in employment.

2.7 Related work

2.7.1 Major related work

Do not use the title of the paper/book/... as the title of the section. Instead summarize what the contribution of this work is in your own words.

Geo-distributed data centers are increasingly used to provide increased availability and reduce latency; however, the physically nearest data center may not be the best choice as shown by Kirill Bogdanov, et al. in their paper "The Nearest Replica Can Be Farther Than You Think" [4]. Exploring decentralized approaches to AI model training, allowing organizations to collaborate on improving tagging accuracy while preserving data privacy.

2.7.2 Major related work

Carrier clouds have been suggested as a way to reduce the delay between the users and the cloud server that is providing them with content. However, there is a question of how to find the available resources in such a carrier cloud. One approach has been to disseminate resource information using an extension to OSPF-TE, see Roozbeh, Sefidcon, and Maguire [5].

2.7.3 Minor related work

Do not use the title of the paper/book/... as the title of the section. Instead summarize what the contribution of this work is in your own words.

2.8 Summary

It is nice to bring this chapter to a close with a summary. For example, you might include a table that summarizes the ideas of others and the advantages and disadvantages of each – so that later you can compare your solution to each of these. This will also help guide you in defining the metrics that you will use for your evaluation.

3 <Engineering-related content, Methodologies and Methods> Use a self-explaining title

The contents and structure of this chapter will change with your choice of methodology and methods. For example, if you have implemented an artifact, what did you do and why? How will your evaluate it.

Describe the engineering-related contents (preferably with models) and the research methodology and methods that are used in the degree project. Give a theoretical description of the scientific or engineering methodology are you going to use and why have you chosen this method. What other methods did you consider and why did you reject them. In this chapter, you describe what engineering-related and scientific skills you are going to apply, such as modeling, analyzing, developing, and evaluating engineering-related and scientific content. The choice of these methods should be appropriate for the problem. Additionally, you should be consciousness of aspects relating to society and ethics (if applicable). The choices should also reflect your goals and what you (or someone else) should be able to do as a result of your solution - which could not be done well before you started. The purpose of this chapter is to provide an overview of the research method used in this thesis. Section 3.1 describes the research process. Section 3.2 details the research paradigm. Section 3.3 focuses on the data collection techniques used for this research. Section 3.4 describes the experimental design. Section 3.5 explains the techniques used to evaluate the reliability and validity of the data collected. Section 3.6 describes the method used for the data analysis. Finally, Section 3.7 describes the framework selected to evaluate xxx.

3.1 Research Process

Image of: steps conducted to do the research Fig: research processes

3.2 Research Paradigm

3.3 Data Collection

(This should also show that you are aware of the social and ethical concerns that might be relevant to your data collection method.)

3.3.1 Sampling

1. Aa 2. Bb 3. Cc

3.3.2 Sample Size

3.3.3 Target Population

3.4 Experimental design/Planned Measurements

3.4.1 Test environment/test bed/model

Describe everything that someone else would need to reproduce your test environment/test bed/model/...

3.4.2 Hardware/Software to be used

3.5 Assessing reliability and validity of the data collected

3.5.1 Reliability

How will you know if your results are reliable?

3.6 Validity

How will you know if your results are valid?

3.7 Planned Data Analysis

3.7.1 Data Analysis Technique

3.7.2 Software Tools

3.8 Evaluation framework

4 [What you did – Choose your own chapter title to describe this]

What have you done? How did you do it? What design decisions did you make? How did what you did help you to meet your goals?

4.1 Hardware/Software design .../ModelSimulation model parameters/...

Figure 4-1 shows a simple icon for a home page. The time to access this page when served will be quantified in a series of experiments. The configurations that have been tested in the test bed are listed in Table 4-1.

Figure 4-1: An example figure in Section.

Column 1	Column 2
Data 1	Data 2
Data 3	Data 4

Table 4.1: An example table in Section.

4-1 is an image 4.1 is a table

4.2 Implementation . . . /Modeling/Simulation/. . .

5 Results and Analysis

In this chapter, we present the results and discuss them.

Keep in mind: How you are going to evaluate what you have done? What are your metrics? Analysis of your data and proposed solution Does this meet the goals which you had when you started?

5.1 Major results

Some statistics of the delay measurements are shown in Table 5-1. The delay has been computed from the time the GET request is received until the response is sent.

Column 1	Column 2
Data 1	Data 2
Data 3	Data 4

Table 5.1: An example table in Section

5.1 is a table

5.2 Reliability Analysis

LALALA

5.3 Validity Analysis

LALALA

5.4 Discussion

6 Conclusions and Future work

«Add text to introduce the subsections of this chapter.»

6.1 Conclusions

Describe the conclusions (reflect on the whole introduction given in Chapter 1). Discuss the positive effects and the drawbacks. Describe the evaluation of the results of the degree project. Did you meet your goals? What insights have you gained? What suggestions can you give to others working in this area? If you had it to do again, what would you have done differently?

6.2 Limitations

What did you find that limited your efforts? What are the limitations of your results?

6.3 Future work

Describe valid future work that you or someone else could or should do. Consider: What you have left undone? What are the next obvious things to be done? What hints can you give to the next person who is going to follow up on your work?

6.4 Reflections

What are the relevant economic, social, environmental, and ethical aspects of your work?

References

Alif, M. A. R. and Hussain, M. (2025). Yolov12: A breakdown of the key architectural features. *arXiv preprint*, arXiv:2502.14740v1. arXiv.org perpetual non-exclusive license.

Angulo, A., Vega-Fernández, J. A., Aguilar-Lobo, L. M., Natraj, S., and Ochoa-Ruiz, G. (2019). *Road Damage Detection Acquisition System Based on Deep Neural Networks for Physical Asset Management*, page 3–14. Springer International Publishing.

Barney, J. (1991). Firm resources and sustained competitive advantage. *Journal of Management*, 17(1):99–120.

Bochkovskiy, A., Wang, C.-Y., and Liao, H.-Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. arXiv:2004.10934v1, 23 Apr 2020.

Civelek, M., Krajčík, V., and Ključnikov, A. (2023). The impacts of dynamic capabilities on smes' digital transformation process: The resource-based view perspective. *Oeconomia Copernicana*, 14(4):1367–1392.

Creswell, J. W. (2014). *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. SAGE Publications, 4th edition.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press.

Hanelt, A., Bohnsack, R., Marz, D., and Antunes Marante, C. (2020). A systematic review of the literature on digital transformation: Insights and implications for strategy and organizational change. *Journal of Management Studies*.

He, Z., Wang, K., Fang, T., Su, L., Chen, R., and Fei, X. (2024). Comprehensive performance evaluation of yolov11, yolov10, yolov9, yolov8 and yolov5 on object detection of power equipment.

Hevner, A. R., March, S. T., Park, J., and Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 28(1):75–105.

Hidayatullah, P., Syakrani, N., Sholahuddin, M. R., Gelar, T., and Tubagus, R. (2025). Yolov8 to yolo11: A comprehensive architecture in-depth comparative review.

Jobin, A., Ienca, M., and Vayena, E. (2019). The global landscape of ai ethics guidelines. *Nature Machine Intelligence*, 1:389–399.

Jocher, G. and Ultralytics (2025). Ultralytics yolov11. https://github.com/ultralytics/ultralytics. Accessed March 2025.

Johnson, R. B. and Onwuegbuzie, A. J. (2004). Mixed methods research: A research paradigm whose time has come. *Educational Researcher*, 33(7):14–26.

Karbouj, B., Topalian-Rivas, G. A., and Krüger, J. (2024). Comparative performance evaluation of one-stage and two-stage object detectors for screw head detection and classification in disassembly processes. *Procedia CIRP*, 122:527–532. 31st CIRP Conference on Life Cycle Engineering (LCE 2024).

Khanam, R. and Hussain, M. (2024). Yolov11: An overview of the key architectural enhancements.

Khanam, R., Hussain, M., Hill, R., and Allen, P. (2024a). A comprehensive review of convolutional neural networks for defect detection in industrial applications. *IEEE Access*, 12:94250–94295.

Khanam, R., Hussain, M., Hill, R., and Allen, P. (2024b). A comprehensive review of convolutional neural networks for defect detection in industrial applications. *IEEE Access*, 12:94250–94295.

Krogh, P. (2009). *The DAM book: Digital asset management for photographers*. O'Reilly, Sebastopol, California, 2nd edition.

Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., Ke, Z., Li, Q., Cheng, M., Nie, W., Li, Y., Zhang, B., Liang, Y., Zhou, L., Xu, X., Chu, X., Wei, X., and Wei, X. (2022). Yolov6: A single-stage object detection framework for industrial applications.

Love, P. E. and Matthews, J. (2019). The 'how' of benefits management for digital technology: From engineering to asset management. *Automation in Construction*, 107:102930.

McCain, E., Mara, N., Van Malssen, K., Carner, D., Reilly, B., Willette, K., Schiefer, S., Askins, J., and Buchanan, S. A. (2021). *Endangered but not too late: The state of digital news preservation*. Donald W. Reynolds Journalism Institute, University of Missouri–Columbia Libraries. OpenAccess. Licensed under CC BY 4.0.

Prince, S. J. (2023). *Understanding Deep Learning*. The MIT Press.

Rane, N. (2023). Yolo and faster r-cnn object detection for smart industry 4.0 and industry 5.0: applications, challenges, and opportunities. *SSRN Electronic Journal*.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788.

Redmon, J. and Farhadi, A. (2016). Yolo9000: Better, faster, stronger.

Redmon, J. and Farhadi, A. (2018). Yolov3: An incremental improvement. arXiv:1804.02767v1.

Sapkota, R., Qureshi, R., Flores-Calero, M., Badgujar, C., Nepal, U., Poulose, A., Zeno, P., Vaddevolu, U. B. P., Khan, S., Shoman, M., Yan, H., and Karkee, M. (2025). Yolo11 to its genesis: A decadal and comprehensive review of the you only look once (yolo) series. *arXiv*, 2406(19407v5).

Soori, M., Arezoo, B., and Dastres, R. (2023). Artificial intelligence, machine learning and deep learning in advanced robotics, a review. *Cognitive Robotics*, 3:54–70.

Teng, X., Wu, Z., and Yang, F. (2022). Impact of the digital transformation of small- and medium-sized listed companies on performance: Based on a cost-benefit analysis framework. *Journal of Mathematics*, 2022:1–15.

Tillväxtverket (2021). Små och medelstora företags digitalisering - vad har betydelse? Technical Report 0366, Tillväxtverket. Accessed: 2025-02-15.

Ultralytics (2020). Comprehensive guide to ultralytics yolov5. Accessed: 21 February 2025.

Ultralytics (2023). Yolov8: A unified architecture for object detection, classification, and segmentation. https://yolov8.com/. Accessed: 2025-03-01.

Ultralytics Inc. (2025). Ultralytics YOLO11. https://docs.ultralytics.com/models/yolo11/. Accessed: 3 March 2025.

United Nations (2015). Transforming our world: The 2030 agenda for sustainable development. Accessed: February 28, 2025.

Verdhan, V. (2021). *Computer Vision Using Deep Learning: Neural Network Architectures with Python and Keras*. Apress, Berkeley, CA, 1st ed. edition.

Vina, A. (2024). The benefits of ultralytics yolo11 being an anchor-free detector. *Ultralytics Blog*.

Wang, A., Chen, H., Liu, L., Chen, K., Lin, Z., Han, J., and Ding, G. (2024a). Yolov10: Real-time end-to-end object detection.

Wang, C.-Y., Bochkovskiy, A., and Liao, H.-Y. M. (2022). Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv preprint arXiv:2207.02696*. Version 1, 6 Jul 2022.

Wang, C.-Y., Yeh, I.-H., and Liao, H.-Y. M. (2024b). Yolov9: Learning what you want to learn using programmable gradient information. *arXiv preprint arXiv:2402.13616*.

Wu, M., Brandhorst, H., Marinescu, m.-c., Moré, J., Hlava, M., and Busch, J. (2022). Automated metadata annotation: What is and is not possible with machine learning. *Data Intelligence*, 5:1–17.

Yin, R. K. (2014). *Case Study Research: Design and Methods*. SAGE Publications, 5th edition.

Zhang, L., Sun, Z., Tao, H., Wang, M., and Yi, W. (2025). Research on mine-personnel helmet detection based on multi-strategy-improved yolov11. *Sensors (Basel, Switzerland)*, 25(1):170–.

Appendices

A Appendix A: Example Appendix Title

This is an example appendix entry. You can include figures, tables, or additional details relevant to your research.

Figure A-1: An example figure in Appendix A.

Column 1	Column 2
Data 1	Data 2
Data 3	Data 4

Table A.1: An example table in Appendix A.

B Appendix B: Another Appendix Example

You can continue adding appendices in a similar manner.

IEEE Editorial Style Manual: