TU WIEN Informatics

Datenorientierte Methoden zur effizienten Montageverifikation mit Deep Learning

Eine Fallstudie in der Herstellung elektronischer Schlösser

BACHELORARBEIT

zur Erlangung des akademischen Grades

Bachelor of Science

im Rahmen des Studiums

Medieninformatik und Visual Computing

eingereicht von

Martin Braunsperger
Matrikelnummer 11909911

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ.Prof. Dipl.-Ing. Dr.techn. Eduard Gröller
Mitwirkung: Univ.Ass. Dipl.-Ing. Daniel Pahr, BSc

Wien, 12. Juni 2025

Martin Braunsperger	Eduard Gröller

TU WIEN Informatics

Data-Centric Methods for Efficient Deep Assembly Verification

A Case Study in electronic Lock Manufacturing

BACHELOR'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Bachelor of Science

in

Media Informatics and Visual Computing

by

Martin Braunsperger
Registration Number 11909911

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Prof. Dipl.-Ing. Dr.techn. Eduard Gröller
Assistance: Univ.Ass. Dipl.-Ing. Daniel Pahr, BSc

Vienna, 12th June, 2025

Martin Braunsperger	Eduard Gröller

Erklärung zur Verfassung der Arbeit

Martin Braunsperger

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 12. Juni 2025

Martin Braunsperger

Danksagung

Ich möchte mich herzlich bei Florian Pauker bedanken, der es mir ermöglicht hat, meine Forschung an der Montagelinie durchzuführen, und ebenso bei Manual Gall und Johannes Cech die mich während meiner Zeit dort betreut haben. Mein Dank gilt auch meinem Betreuer Daniel Pahr und meinem Professor Eduard Gröller für ihre Unterstützung und dafür, dass sie mir die Zeit gelassen haben, diese Arbeit trotz meines langsamen Fortschritts abzuschließen. Ganz besonders möchte ich meiner Familie, meinen Freunden und meinen Kommilitoninnen und Kommilitonen für ihre Unterstützung danken. Ein riesiges Dankeschön geht an Gerald Kortschak und meinen Vater, die immer ein offenes Ohr für meine Ideen und Probleme hatten und mir mit Rat zur Seite standen.

Acknowledgements

I would like to express my sincere gratitude to Florian Pauker for providing me with the valuable opportunity to conduct my research at the assembly line at EVVA, and to Manuel Gall and Johannes Cech for their guidance throughout my time there. I am also deeply thankful to my supervisor and professor for their patient guidance and for granting me the necessary time to complete this thesis, despite my slow progress. Finally, I extend my thanks to my family, friends, and fellow students for their support. A special thank you goes to Gerald Kortschak and my father for their patient ear, insightful advice, and for patiently listening to my endless ramblings and challenges during this process.

Kurzfassung

Diese Arbeit untersucht dateneffiziente Deep Learning Methoden zur visuellen Montageverifikation in einer Produktionslinie für hochgradig anpassbare elektronische Schlösser. Im Mittelpunkt stehen drei datenorientierte Ansätze: Der erste Ansatz nutzt die bekannte statische Geometrie des Werkstückträgers, um die Objekterkennung auf eine Reihe Klassifikationen auf vordefinierten Bildbereichen zu reduzieren (ROI-basierte Klassifikation). Der zweite Ansatz erzeugt automatisch Pseudo-Annotationen für die Objekterkennung, indem Bildbeschrifungen mit den beschränkten Bauteilpositionen kombiniert werden. Der dritte Ansatz erweitert den Datensatz durch synthetisch generierte Bilder, die durch die Kombiniation segmentierter Bauteile mit Hintergrundbildern erzeugt werden.

Ziel der Arbeit ist es, das Potenzial dieser Ansätze zur Reduktion des manuellen Annotierungsaufwands zu bewerten. Hierzu werden alle Methoden unter realen Produktionsbedingungen systematisch verglichen und hinsichtlich ihrer Stärken, Schwächen und praktischen Einsatzmöglichkeiten analysiert. Die Ergebnisse zeigen, dass durch eine gezielte Nutzung dieser domänenspezifischen Strukturen eine robuste Modellleistung auch mit deutlich reduziertem Annotierungsaufwand erreicht werden kann.

Abstract

This thesis investigates data-efficient deep learning methods for visual assembly verification in a highly customizable electronic lock production line. We examine three data-centric strategies. First, ROI-based methods leverage the fixed geometry of the workpiece carriers by reformulating object detection as a set of classification or regression tasks over predefined regions. Second, pseudo bounding boxes are created by combining image-level labels with known part positions to automatically generate object detection annotations without manual labelling. Third, synthetic training data is produced by compositing cropped part images with background scenes, thereby increasing dataset diversity and reducing the need for extensive manual data collection.

These methods are evaluated on a real-world dataset collected during regular production to assess their effectiveness in reducing manual annotation effort. We provide a comprehensive comparison of data-centric approaches, highlighting their respective strengths and limitations. The results demonstrate that leveraging the structured nature of the assembly environment enables accurate model performance with substantially reduced annotation requirements.

Contents

CHAPTER 1

Introduction

1.1 Motivation

Visual inspection plays a critical role in manufacturing, ensuring consistent product quality and adherence to specifications. It is particularly important in manual assembly lines, which are prone to human error due to the unsuitability of our cognitive system for repetitive, monotonous tasks [Arn97]. Automated assembly processes face their own set of challenges, including calibration issues, machine wear, and software glitches. As a result, quality control has remained a staple since the early days of industrial production. However, manual inspection encounters similar issues to manual assembly lines [DS83; KV15].

As such, automated quality control systems based on traditional machine learning techniques emerged. Given the narrow application domain and controlled environments, it was possible to achieve satisfactory results based on handcrafted features [Wan+18]. However, these traditional methods were labour-intensive to develop, sensitive to environmental changes, and often struggled to adapt to new or varying product designs [BCV13; GBC16; NS21]. Consequently, attention has shifted towards deep learning, which has achieved remarkable results in many areas since its resurgence in 2012 [KSH12; Kon+20; Geh+17]. With strong generalization, the ability to operate effectively in less controlled environments, and reduced need for manual feature engineering [KCT20], deep learning has become a cornerstone of modern visual inspection systems. However, its effectiveness typically depends on access to large and diverse datasets, which can be an issue in many scenarios due to the specialized nature of the applications.

Visual inspection tasks generally fall into two categories: defect detection and assembly verification [Hüt+24] [1]. Defect detection aims to identify surface flaws (e.g., scratches, cracks) and structural defects (e.g., holes, cavities) [Czi+20; Yan+20; JB23]. It dominates

[1] A more extensive classification will be introduced in Section 2.1.

current research due to its relevance in high-volume, standardized manufacturing [Hüt+24; JB23; Lee+23; Che+23a; Tao+22]. In contrast, assembly verification, the focus of this thesis, confirms that all required components are present and correctly assembled [Hüt+24; RL18]. It is especially relevant in the context of configurable or customizable products, particularly as demand for personalization grows. Assembly verification remains under-explored, largely due to the lack of publicly available datasets, and the dependence on proprietary data. These factors limit accessibility for researchers as well as hinder reproducibility, and comparison across studies [Hüt+24; KCT20]. Additional challenges include the high costs of data collection and annotation, as well as significant class imbalance across different variants and configurations.

Despite these challenges, the structured and controlled nature of assembly environments offers distinct advantages. The placement of individual components is typically constrained, such as by a workpiece carrier or by being mounted on a printed circuit board (PCB). Components of the same type also tend to exhibit a highly uniform appearance. These characteristics provide an opportunity to reduce data requirements and improve model performance. As manufacturing continues to shift toward mass customization[2], the increase in product variety makes data efficiency even more critical for ensuring reliable and scalable assembly verification systems.

1.2 Objective

This thesis explores data-efficient deep learning techniques for visual assembly verification, with a focus on minimizing annotation effort. The core idea is to fully utilize the specific characteristics of the assembly environment through data-centric strategies.

We strive to answer the following research questions:

- **How can constrained object locations and low intra-class variance reduce dataset requirements?**
- **What is the trade-off between increased dataset size and class imbalance in assembly verification tasks?**
- **Can synthetic or weakly labelled data be used to reduce the need for manual annotations without compromising model performance?**

To answer these questions, three principal approaches are examined:

- **ROI-based methods** (Chapter 4) leverage the spatial layout of the workpiece carrier to reformulate the verification task as a set of independent classification or regression problems.

[2] Mass customization is a strategy that involves customers in the manufacturing or assembly process, delivering customized products at a price comparable to mass-produced items [KH06].

- **Pseudo-annotations** (Section 5.1) replace manually annotated bounding boxes with pseudo-annotations automatically derived from weak supervision.
- **Synthetic data generation** (Section 5.2) uses a custom image composition pipeline to increase effective dataset size by increasing diversity and reducing imbalance.

Together, these approaches are evaluated to understand how annotation effort can be reduced without sacrificing performance. Finally, we examine the trade-off between constructing datasets from manually assembled examples and capturing them passively during production.

1.3 Use Case

The research is based on a case study conducted at an electronic lock assembly line of EVVA in Vienna. It produces a wide range of highly customizable locking systems, with variations in length, locking mechanisms, and surface finishes, among other options, resulting in significant product variability. An example of such a lock system can be seen in Figure 1.1. Further details on the types of locks and possible configurations can be found in Section 3.3. The use case is uniquely suitable, as a prior feasibility study by an external contractor failed to achieve satisfactory results using traditional machine learning methods.

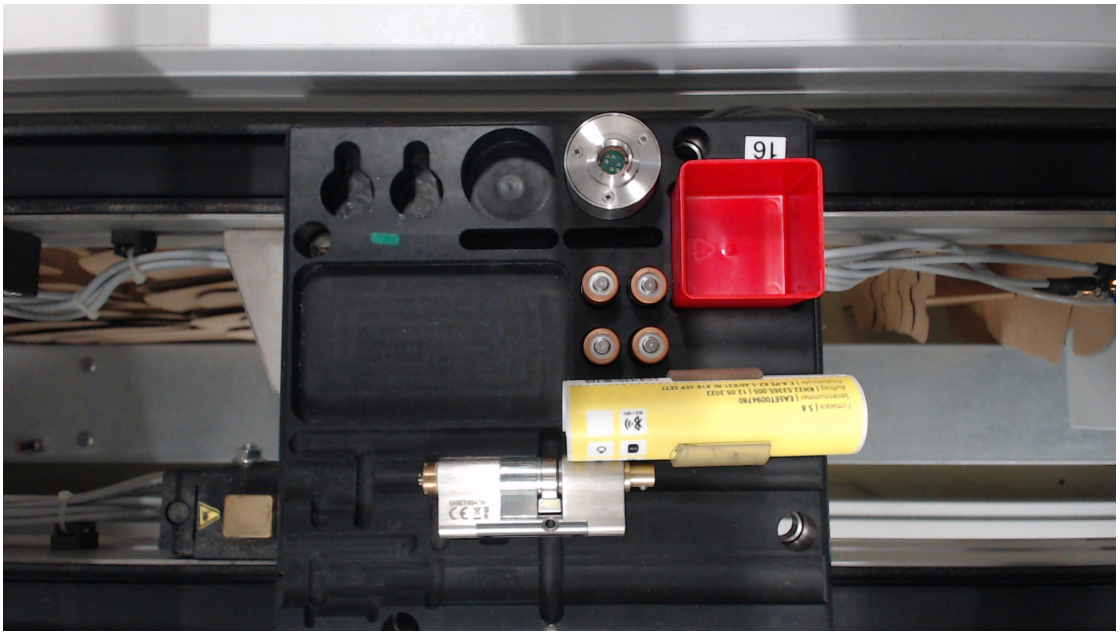


Figure 1.1: An example of a workpiece carrier on the assembly line holding a partially assembled lock system.

CHAPTER 2

Related Work

Given the limited research directly addressing data-efficient deep learning for assembly verification, this chapter instead discusses the key areas relevant to this thesis. We begin by situating assembly verification within the broader context of visual quality control. We then review general strategies for improving data efficiency in deep learning. Finally, we provide a brief overview of methods for synthetic data generation.

2.1 Visual Quality Control

Visual quality control in production lines encompasses a broad range of use cases, each with unique requirements and quality criteria [KCT20]. To contextualize our use case, we consider applications along a spectrum based on how specific the failure cases are defined, as illustrated in Figure 2.1. At one end lies assembly verification, the focus of this thesis, where the errors are clearly defined, such as missing, misplaced, or incorrect components (e.g., a wrong screw or missing battery). At the other end is anomaly detection, which aims to identify unexpected patterns or events (e.g., foreign objects or damage). This categorization helps clarify how different tasks place different demands on annotation, supervision, and generalization capability.

2.1.1 Anomaly Detection

Anomaly detection aims to identify outliers during production, which often indicate defects, missing components, foreign objects, or other unwanted occurrences [KD19; Maz+20]. Because the goal is to detect any type of failure, it is infeasible to collect a comprehensive dataset of all possible failure cases. As a result, anomaly detection typically relies on large sets of unlabelled data, such as through semi-supervised and self-supervised methods. These techniques learn what constitutes correct products and operating conditions from the training distribution, even if the majority of images are

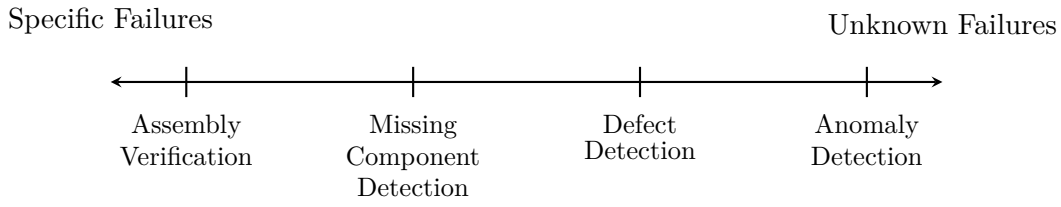


Figure 2.1: Spectrum of visual quality control tasks categorized by the specificity of failure cases. Tasks range from well-defined issues such as those in assembly verification to open-ended problems encountered in anomaly detection.

unlabelled [JB23; Tao+22; Maz+20]. Prominent approaches also include autoencoders (AEs) [MTM22; UYY21; KB20], generative adversarial networks (GANs) [BEM21; RMM20], and normalizing flows (NFs) [MTM22].

2.1.2 Defect Detection

Defect detection involves identifying specific surface or structural flaws, such as scratches, dents, cracks, holes, or soldering issues[Czi+20; Yan+20; JB23; Mig20]. In contrast to anomaly detection, the types of defects are often predefined, enabling the use of supervised methods like object detection or segmentation [Hüt+24; JB23; HWZ19; Al+22; Maz+20]. Detection strategies range from multi-class classification of distinct defect types to simpler binary classification distinguishing defective from non-defective instances [Hüt+24; Ben+21; Yan+19; SZ23]. Although the location of defects is generally unconstrained, certain applications allow for the use of ROI-based classification due to the fixed positions of components. An example of this is the identification of soldering defects on PCBs [Met+19; Mig20]. A key challenge in defect detection is the high cost and effort involved in producing detailed annotations. To address this, weakly-supervised and semi-supervised methods are increasingly used, as they can learn from image-level labels and large pools of unlabelled data respectively. Readily available public datasets further help reduce dataset requirements. Additionally, data augmentation and synthetic data generation are commonly applied to expand training diversity and improve model generalization [JB23; Tao+22; Maz+20].

2.1.3 Assembly Verification

Assembly verification, the subject of this thesis, ensures that products meet both quality standards and order specifications. This typically includes verifying that all required parts are present and conform to the customer's order in terms of visual properties such as dimensions and surface finish [Hüt+24; Sta+23]. A common example is the inspection of PCBs to confirm correct placement of capacitors, resistors, and integrated circuits (ICs) before soldering [Ara+24; Kur+20].

The majority of recent work relies on object detection, either directly, treating each component variant as a separate class, or in two-stage methods, where a detector first

identifies broadly categorized parts and a second model performs finer classification. Many industrial settings necessitate an object detection-based approach, such as in domains like aerospace and automotive assembly for verifying fasteners (e.g., screws, rivets, joints) [DLZ24; AAJ24; Zha+21b]. However, a prominent portion of use cases involve constrained component layouts, where parts occupy fixed or predictable locations. In these settings, ROI-based classification offers a more efficient alternative by applying classifiers to predefined regions rather than performing full detection. To our knowledge, only Lim et al. [LKP19] apply this method, limited to highly structured PCBs and without comparison to object detection methods. This thesis considers ROI-based classification in a broader context, and compares it against standard object detection-based approaches.

Recent works in assembly verification are summarized in Table 2.1. Most approaches employ transfer learning, typically starting with weights pretrained on ImageNet [Den+09], and favour lightweight single-stage detectors like YOLO [TC23] and SSD [Liu+16] for deployment efficiency. Backbone architectures such as MobileNet [How+17], EfficientNet [TL19], AlexNet [KSH12], and shallow custom CNNs are common due to hardware constraints in deployments.

Ref.	**Detection**		**Metric**	**Result**
[Sha+24]	YOLOv8s		Accuracy	0.98
[OK19]	ACF (AlexNet)		Accuracy	0.9722
[Kur+20]	Faster R-CNN (AlexNet)		Accuracy	0.883 - 0.9994
[Maz+20]	SSD (MobileNet)		Accuracy	0.7763
[Liu+19]	Mask R-CNN (ResNet-101)		Accuracy	0.937
[Ara+24]	Custom YOLO-like (MobileNet)		AP	0.9997
Ref.	**Classification**		**Metric**	**Result**
[LKP19]	Custom FCN + Custom Classification		Accuracy	0.98
Ref.	**Detection**	**Classification**	**Metric**	**Result**
[Zha+24b]	Hough Transform	ResNet-34	Accuracy	0.997
[Sta+23]	YOLOv5s	EfficientNet	AP_{50}	0.993 (wheels)
			Accuracy	0.9872 (rims)

Table 2.1: Models used in recent literature for assembly verification

The datasets used in recent assembly verification research (Table 2.2) vary significantly in size and scope, but most are custom-built and contain hundreds to thousands of images. Only two works operate with notably small datasets: Liu et al. [Liu+19] use just 64 training images, but compensate with densely populated scenes and pixel-level segmentation masks. Aras et al. [Ara+24], in contrast, rely on only 27 original images, but focus exclusively on a single class. As such, neither represents a general approach to learning from limited, real-world data across multiple classes. Notably, all other datasets with more than 10 classes include over one thousand training images, which highlights a

lack of methods explicitly designed for low-data regimes. Additionally, the absence of standardized benchmarks and inconsistencies in data formatting hinder reproducibility and cross-study comparison. This thesis addresses these gaps by evaluating performance under constrained data conditions and providing a direct comparison between object detection and ROI-based classification approaches.

Ref.	Classes	Content	Training Size	Full Size	Notes
[Sha+24]	3	Unnamed (base, spring, piston, cap)	268	338	balanced dataset
[OK19]	2	Wall clock	320	356	balanced dataset
[Kur+20]	3	ICs on PCB	$\approx 1,000$	unknown	
[Ara+24]	1	ICs on PCB	12,000	15,000	27 original images
[LKP19]	14	SMCs	7,659	12,481	
[Zha+24b]	22	Saw Chains	3,094	4,420	
[Sta+23]	31	Car wheels	1,000	1,500	
		Wheel rim	2,100	3,300	
		Wheel bolts	400	600	
[Maz+20]	6	Brake disc and calliper	unknown	321	
[Liu+19]	≥ 14	Computer chassis ports	64	80	

Table 2.2: Datasets used in recent literature for assembly verification

2.2 Data-Efficient Learning

In addition to task-specific strategies for structured inspection, various general-purpose approaches have been developed to reduce the annotation demands of deep learning. However, most of these are poorly suited to this use case, as described in this section and verified in Section A.2.

2.2.1 Transfer Learning

A common approach is to fine-tune models pretrained on larger datasets. Even if the source and target domains differ, this nevertheless often improves performance or at least convergence speed [PG22]. TURTLE [GJB24] goes further by requiring no labels through clustering of feature representations from a pretrained model using only the number of target classes as input.

2.2.2 Meta-Learning

Meta-learning, or "learning to learn", enables models to quickly adapt to new tasks with minimal data by leveraging knowledge acquired from a variety of previous tasks [Gha+24]. Context-Aware Meta-Learning (CAML) [Fif+24], for instance, reframes visual

meta-learning as a sequence modelling task, allowing a fixed pretrained feature extractor to adapt to new visual tasks without fine-tuning.

2.2.3 Few-Shot Learning

Few-shot learning aims to train models that can generalize from just a few labelled examples. Despite its theoretical appeal, few-shot learning is often outperformed by simple supervised approaches, such as those examined in this thesis [NH19; Che+19].

2.2.4 Self-Supervised Learning

Self-supervised learning uses auxiliary tasks that require no human labels, such as predicting masked regions [He+22] or distinguishing between augmented views of the same image [Che+20]. While promising in reducing annotation effort, these methods typically require large and diverse datasets to learn meaningful representations. This makes them ill-suited for our objective of distinguishing fine-grained, visually similar part variants with limited data.

2.2.5 Weakly-Supervised Object Detection

Weakly-Supervised Object Detection (WSOD) involves training detectors using only image-level labels, without the need for manual bounding boxes [Zha+21a]. Common approaches include:

Multiple Instance Learning (MIL) [Car+18]: Treats each image as a bag of candidate regions (instances), and learns to identify which region is responsible for the image-level label. This allows the model to localize objects without explicit bounding boxes.

Class Activation Maps (CAM) [Sel+20]: Use the feature maps of trained classifiers to highlight image regions most relevant to a given class. These heatmaps can be refined into pseudo-bounding boxes for object localization.

While effective in some domains, WSOD appears ill-suited for structured assembly tasks with tightly packed, small, or visually similar components, especially in comparison to the high-quality pseudo bounding boxes that can be directly generated from known part positions, as described in Section 5.1. WSOD could, however, be used to refine these pseudo annotations.

2.2.6 Semi-Supervised Detection

Semi-supervised detection uses a small set of labelled images in conjunction with a larger pool of unlabelled ones. Pseudo-labels are generated on the unlabelled data to guide further training. A common method is Mean Teacher [TV17], where a teacher model produces predictions that a student model learns to match.

In the context of lock assembly, this approach faces challenges. Small, tightly packed, and visually similar components make reliable pseudo-labelling difficult, similar to the challenges seen with WSOD. In particular, the high dataset imbalance means that semi-supervised methods appear to offer limited benefit in this use case.

2.2.7 ROI-Based Classification

ROI-based classification reduces detection to a set of classification tasks over predefined image regions, making it well-suited for structured setups where part locations are fixed or predictable. This enables simpler models, faster training, and easier annotation, often requiring only image-level labels. However, limited context and background variation can lead to overfitting and reduced generalization. It also lacks the spatial precision of bounding box or segmentation-based methods. Despite its advantages in constrained settings, ROI-based classification is rarely used in recent literature, with only two examples: Lim et al. [LKP19] apply it to PCB component classification and Miguel [Mig20] uses it for solder defect detection in electric toothbrushes. Critically, both of these applications involve fully fixed positions. This thesis, in contrast, applies ROI-based classification to a more challenging scenario and additionally incorporates regression tasks.

2.3 Synthetic Data Generation

Deep learning's success relies heavily on large, diverse datasets. In specialized tasks like assembly verification, public datasets are rare, and collecting a custom dataset can be time-consuming and expensive. Basic augmentation techniques (e.g., rotation, mirroring, color jitter) help increase diversity, but they fall short of replicating all real-world variation [Nik21]. Synthetic dataset generation offers a scalable alternative, enabling the creation of balanced datasets with automatically generated, error-free labels [Son+24b; Wan+23].

Three main synthetic data generation approaches have emerged [Nik21]:

CGI-based generation uses 3D models and game engines to render synthetic images. This offers complete control over the composition and enables perfect automatic labelling [Tan+21], but requires extensive setup and access to accurate object models. Bridging the domain gap to real-world data also remains a key challenge [Nik21]. This method is employed by Tang et al. [Tan+21], who use Unity3D to generate images of a complex aero-engine and its components.

Generative methods, such as Variational Autoencoders (VAEs), Generative Adversarial Networks (GANs), and diffusion models learn to synthesize realistic images from the dataset distribution. While these methods are effective for common objects, thanks to large-scale public datasets, they often struggle to generalize to niche domains without sufficient data. Li et al. [LXJ23] apply GANs to simulate PCB

defects, while He et al. [He+23] as well as Yu and He [YH22] generate defect images from scratch.

Compositing combines segmented parts with varied backgrounds to cheaply generate new samples [Zha+24a]. This approach can be enhanced by hybridizing with CGI or generative techniques [Son+24a]. Compositing is particularly effective with limited data, allowing for the creation of diverse configurations and balanced datasets with minimal initial effort. Aras et al. [Ara+24] paste IC cutouts or blank green rectangles onto PCB images.

Among the three approaches, compositing is the most practical for assembly verification. It requires minimal setup and is well-suited to structured environments with consistent part placement. CGI-based methods offer precise control and perfect labels, but require time-consuming setup and are better suited for occlusion-heavy tasks or for generating hard-to-obtain ground truths. Generative models demand substantial datasets in the target domain and can struggle with the complex configuration and imbalanced datasets found in assembly verification. While generating cropped components with limited data is feasible, it is only useful if intra-class variance is high. Aras et al. [Ara+24] demonstrate the potential of compositing with IC overlays on PCBs, though their approach remains limited in both scope and sophistication. This thesis extends that idea with a more targeted and robust compositing pipeline for lock assembly.

2.4 Summary

Current research in assembly verification largely focuses on object detection, with limited exploration of data-efficient alternatives. ROI-based classification has shown potential but remains under-explored, with only a handful of examples in the literature and no comparison benchmark. Existing synthetic data methods also often rely on trivial compositing strategies, such as those used by Aras et al. [Ara+24], which limit realism and generalization.

This thesis addresses these gaps by:

- Applying ROI-based classification to a more challenging, less constrained use case.
- Introducing a pseudo-annotation approach based on the constrained component positioning.
- Improving upon the simplistic compositing strategies for synthetic data generation to enhance applicability.
- Providing a unified benchmark to systematically compare these approaches under consistent conditions.

CHAPTER 3

Dataset

This chapter introduces the dataset used in this thesis. It was collected directly from an electronic lock manufacturing line and reflects real-world conditions, including class imbalance and product variability. We begin with the data acquisition setup, followed by the preprocessing steps, and finally we detail the dataset composition, covering the components and configurations relevant to this study.

3.1 Data Acquisition

The images for this dataset were gathered from the electronic lock manufacturing line at EVVA Sicherheitstechnologie GmbH in Vienna. This production line assembles electronic locking systems, such as Airkey and Xesar, across 8 sequential stations. A camera positioned above the conveyor belt automatically captures images of carriers holding the lock body, thumbturns, and other components just before packaging. The majority of the dataset was collected continuously during regular production, accurately reflecting the real-world product composition. However, parts or variants that appeared very infrequently were excluded if they had fewer than 15 instances or were supplemented with additional hand-picked or manually captured production images. Additionally, a large number of extra images were collected for thumbturns with missing screws, as this important failure mode was not otherwise captured in the production data. While we expect the findings of this thesis to generalize to the excluded classes, their low real-world sample count makes a reliable evaluation infeasible.

The images for the dataset were gathered using a webcam positioned at the second-to-last station above the conveyor belt, thus seeing the almost fully assembled lock system just before packaging. Whenever a carrier enters the station, a signal is sent to trigger the webcam, which then captures an image. The full setup at the station is depicted in Figure 3.1 with one of the raw collected images shown in Figure 3.2.

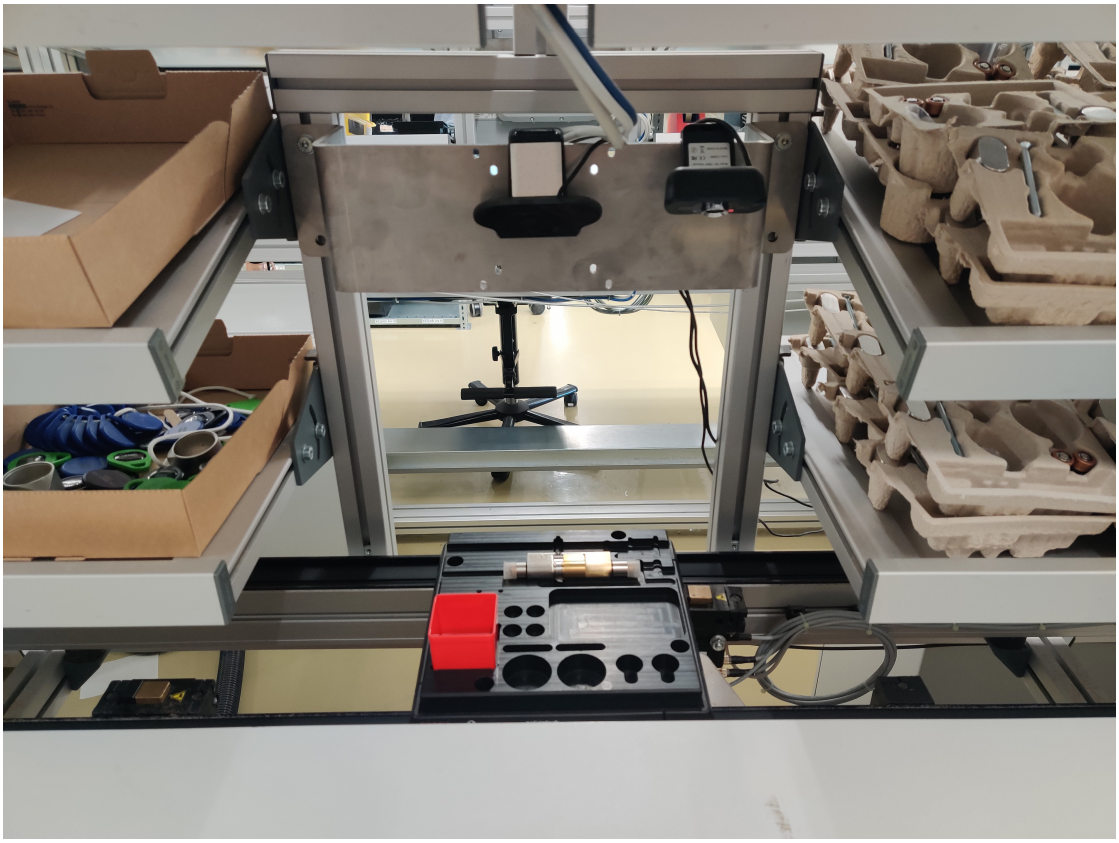


Figure 3.1: The experimental data acquisition setup at EVVA Sicherheitstechnologie GmbH in Vienna.

The bulk of the dataset was collected passively during production, with images automatically captured as products move along the assembly line. This approach ensures that the dataset reflects real-world production conditions without requiring additional effort beyond the initial setup. However, it introduces a significant imbalance, as certain product variants appear far more frequently than others due to differences in their popularity and production frequency.

3.2 Preprocessing

Collected images were filtered to exclude those that are drastically overexposed, significantly occluded or feature moving carriers. These issues arise from the temporary and experimental nature of the data collection setup, and are unlikely to occur in a proper deployment. They generally also provide little training value, as they either lack salient features or have ambiguous ground truth annotations. Additionally, images with misplaced parts are set aside for a separate evaluation dataset examined in Section 7.2.3.

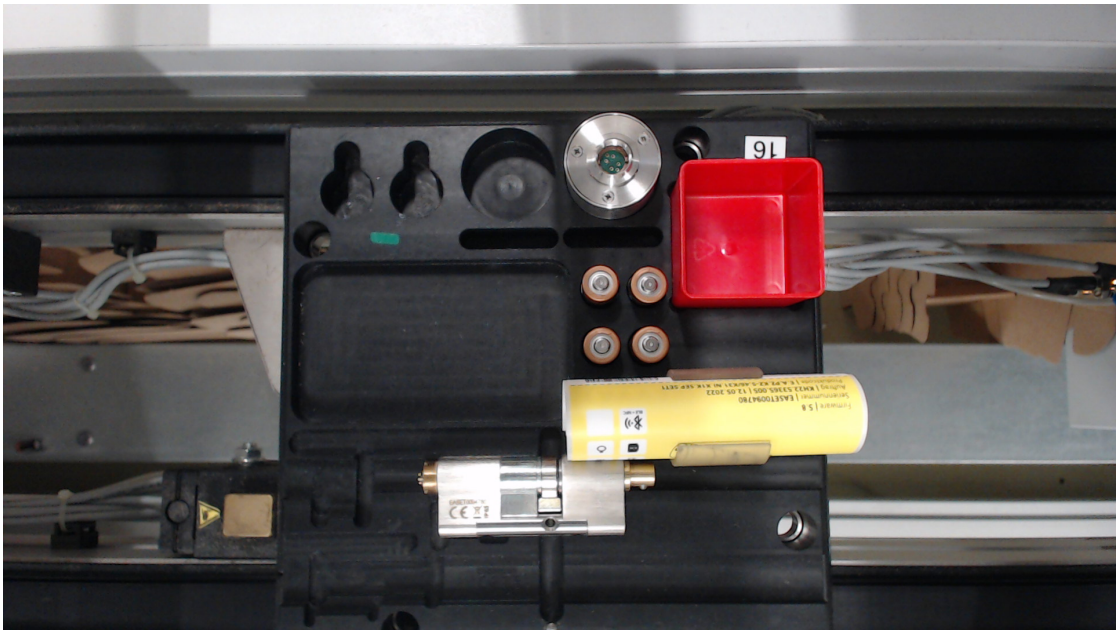
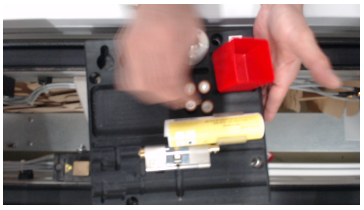


Figure 3.2: Example of an image from the dataset, captured by the overhead camera during production.

(a) Overexposed image



(b) Occluded and moving carrier

(c) Carrier with misplaced parts

Figure 3.3: Example images excluded from the dataset.

After filtering, the corners of each carrier are located in the images. The interior region is then cropped and warped into a rectangular shape as shown in Figure 3.4. This eliminates irrelevant features outside the carrier region, homogenizes variations in camera positioning, and aligns the carrier and its components with the pixel grid, thus removing the need for oriented bounding boxes in Chapter 5. The final image size is about 995×893 pixels with small variations due to the perspective corrections.

3.3 Composition

This section introduces the various configurations and customizations available for the cylinders in the manufacturing process. A wide range of lock profiles are produced on the same manufacturing line according to the varying standards in different countries. Euro-

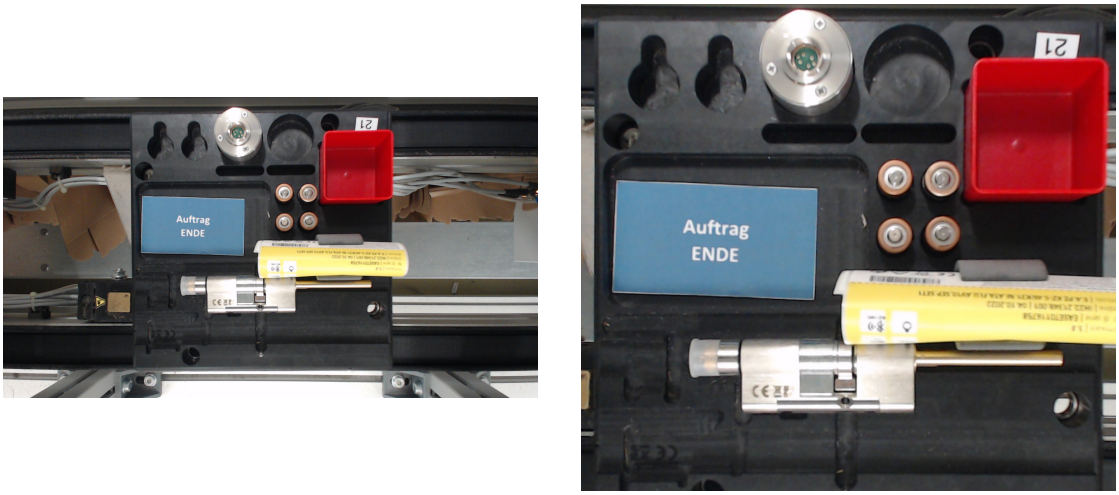


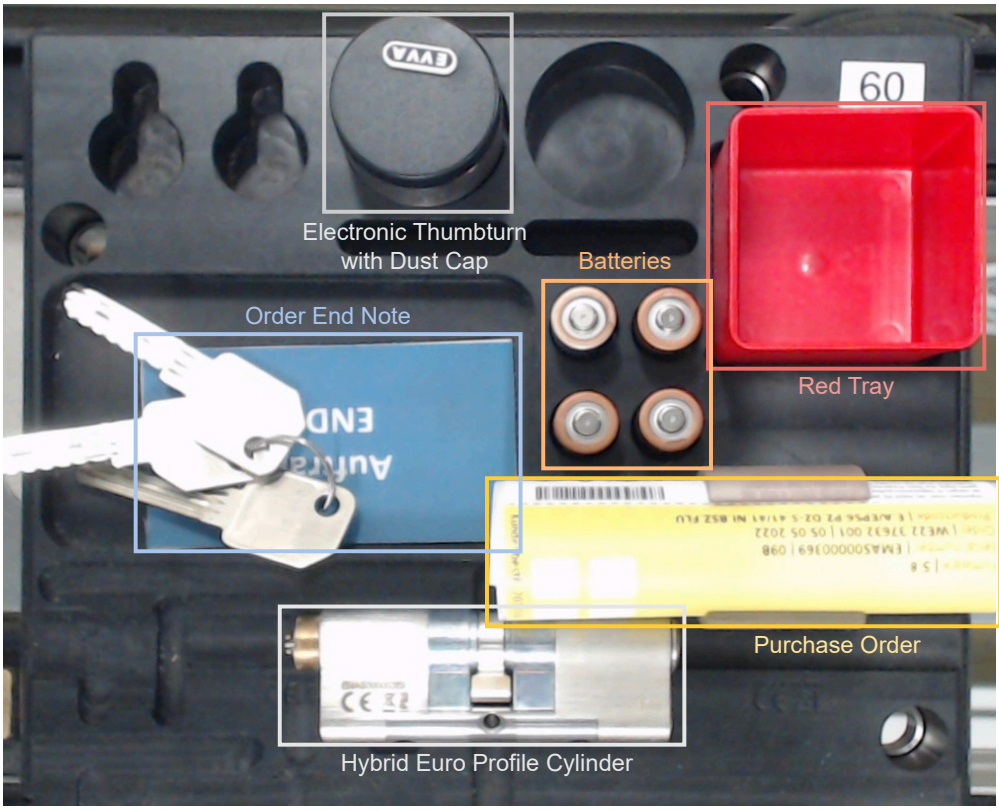
Figure 3.4: Example of dataset preprocessing with the original image on the left and processed image on the right.

and Swiss round cylinders can be either one-sided or double-sided and have configurable lengths in 5 mm increments. They can have different cylinder ends, such as a keyway, an attachment point for an electronic or mechanical thumbturn, an adaptable thumbturn axis, a blind module (a solid, closed end), or a blind module with an anti-panic function. The electronic thumbturn plug can be protected by a dust cap and can optionally feature a length extension. The locking mechanism can be either a standard cam, a cogwheel, or absent. The SEP cam has a different appearance for one-sided and double-sided cylinders. Additionally, the lock's surface finish can be customized as Nickel, Polished Brass, or Black Patina, depending on the desired appearance. Scandinavian oval cylinders and cam lock cylinders do not have visible customization, except that the Scandinavian profile can be either an inside (SKI) or outside (SKA) variant.

Besides the cylinder bodies, one electronic thumbturn should be present for each corresponding cylinder attachment point. Additionally, there is a yellow paper containing the purchase order, a small tray for holding extra parts, and a set of notes containing information for the line workers about the order. The full list of components and properties considered in this thesis is detailed in Table 3.1 with some example images shown in Figure 3.5. Notably, some exceedingly rare variants, with fewer than 10 instances over the collection period of a few months, have been excluded from the case study.

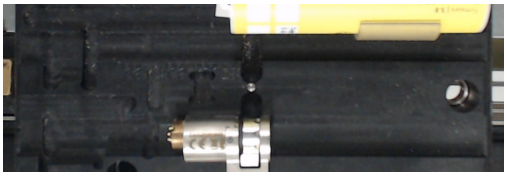
Part	Property	Values
Euro Profile Cylinder, Swiss Round Profile Cylinder	Presence	Yes/No
	Type	One-/Double-Sided
	Length (Left, Right)	27, 31, 36, 41, ...
	Endpiece (Left, Right)	Electronic thumbturn, Keyway, X1K (Inside mechanical thumbturn), ATA (Adaptable thumbturn axis), BLIND (Blind module), BLIND-FAP (Blind module with anti-panic function)
	Elongated outside thumbturn axis (Left, Right)	AV05 (5 mm), AV10 (10 mm), AV15 (15 mm) only available with electronic thumbturn endpiece
	Surface Finish (Left, Right)	Nickel, Polished Brass, Black Patina
	Cam	SEP (different versions for single and double cylinders), Cogwheel
Oval Scandinavian Cylinder	Presence	Yes/No
	Type	SKA (External) SKI (Internal)
Cam Lock Cylinder	Presence	Yes/No
Electronic Thumbturn	Presence (Left, Right)	Yes/No
	Dust cap (Left, Right)	Present, Absent
	Screws (Left, Right)	0, 1, 2, 3 (Only visible without cap)
Padlock	Presence	Yes/No
Red Tray	Presence	Yes/No
	Has Content	Yes/No
Battery	Presence	Yes/No
Order Notes	Presence	Yes/No
	Type	Order-End, Bulk-Packaging, Flu, Xesar, Airkey

Table 3.1: Parts and properties to be detected in this case study

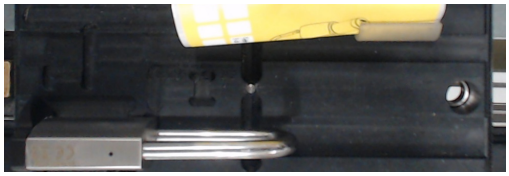


(a) Double-sided hybrid Euro cylinder with electronic thumbturn and keyway

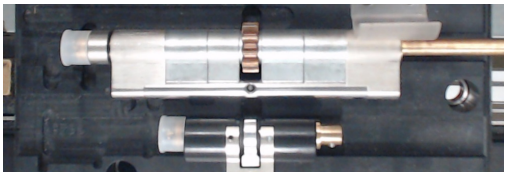
(b) External Scandinavian oval cylinder



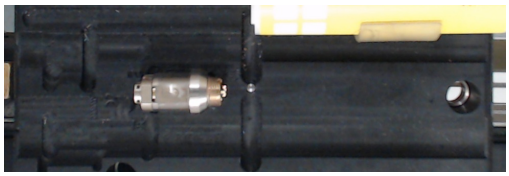
(c) One-sided Swiss round cylinder



(d) Padlock



(e) Top: Long Euro cylinder with cogwheel cam, extended electronic thumbturn attachment point and adaptable thumbturn axis (ATA).
Bottom: Swiss round cylinder with Black Patina finish, standard SEP cam and both electronic and mechanical thumbturn attachment points.



(f) Small cam lock cylinder

Figure 3.5: Dataset excerpt showcasing various lock types and configuration variants.

CHAPTER 4

ROI-Based Methods

Object detection is the most widely used approach for automated assembly verification, enabling both localization and classification of components. However, it typically requires time-consuming bounding box annotations, which can present a significant bottleneck. To mitigate the issue, this chapter explores a method that leverages the compartmentalization of the workpiece carrier. By splitting the image into predefined regions of interest (ROIs), each corresponding to a specific component, the detection task can be decomposed into a set of simple, localized classification or regression problems. The ROIs chosen for this case study are depicted in Figure 4.1.

Since part positions are constrained, each ROI covers only a narrow and consistent input domain. This simplifies the task and allows for the use of smaller, more data-efficient models. Unlike object detection, this method only requires image-level labels for training, reducing annotation time by a factor of 10 to 20 for this dataset. When the assembly line can track expected products passing through, these labels can be automatically inferred and assigned to the corresponding regions with minimal human intervention.

A key limitation of this approach, however, is its inability to detect misplaced components. This limitation is inherent to the approach, with its severity depending on the frequency of such misplacements. As an alternative, Chapter 5 will examine object detection-based approaches.

This chapter outlines a categorization of regions, the construction of per-region datasets, as well as the used model architecture, and training strategies. Considerations for evaluation and their results can be found in Chapter 6 and Section 7.1 respectively.

4.1 Types of Regions

The workpiece carrier holds many different kinds of components, each subject to its own set of constraints. Notably, some components do not always occupy a fixed position,

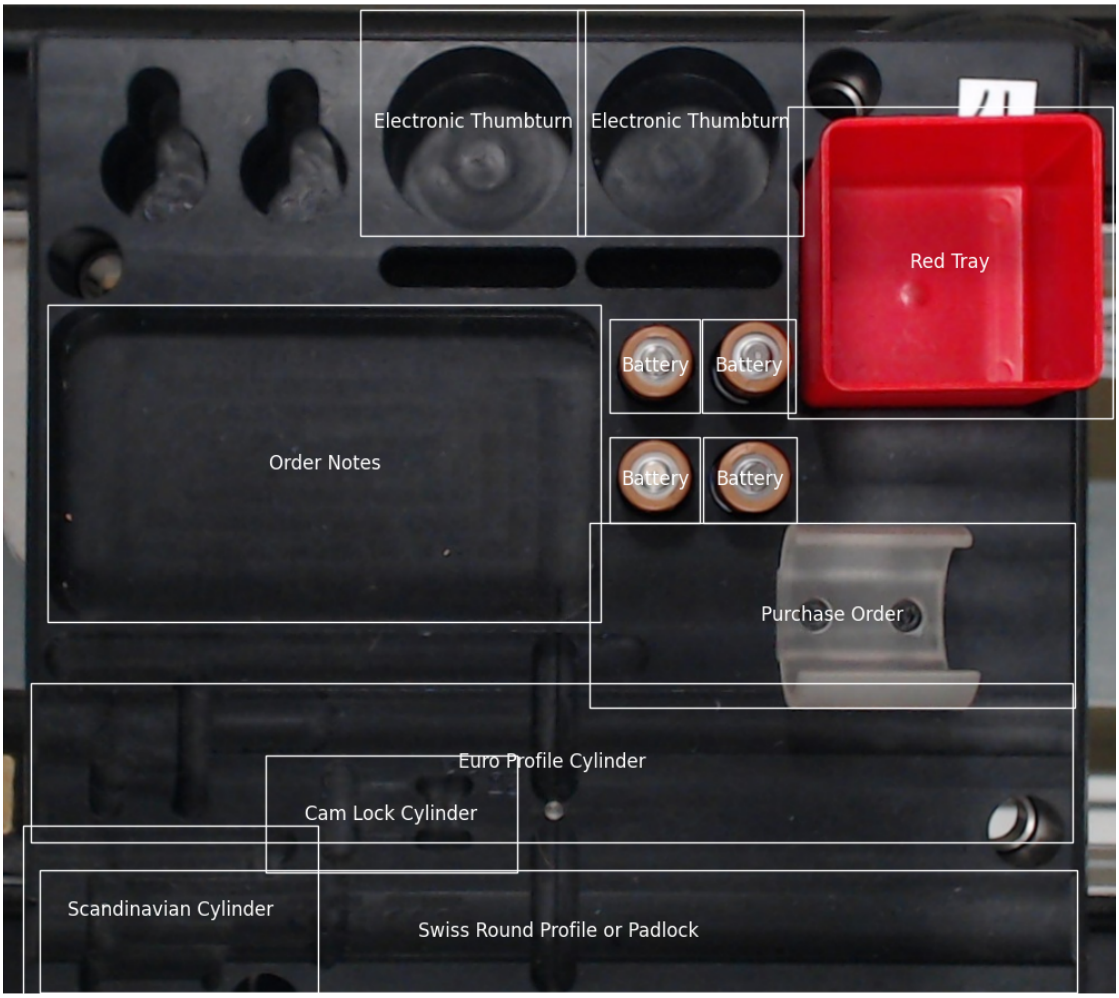


Figure 4.1: An empty workpiece carrier with the applied ROIs overlaid. The regions for Euro and Swiss round profile cylinders are further subdivided into left, right and centre.

some regions may contain multiple parts, and some components are positioned relative to other components on the carrier. In addition, each part has distinct relevant properties to detect, necessitating slightly different methods for each region. To address this, we classify regions based on two main factors: the spatial constraints of the components within and the specific tasks required for their identification (see Table 4.1).

4.1.1 Spatial Constraints

Regarding spatial constraints, regions fall into three distinct categories:

Tightly Constrained These regions correspond to custom indentations designed to hold specific components in a fixed position. Since the object's location does not vary, the camera consistently captures the component from the same angle and perspective. This results in the lowest visual variance and data requirements.

Lightly Constrained In these regions, components are not confined to an exact location, but instead reside within a larger space. Their orientation or location may vary slightly across different images.

Relative Some regions are not independent, but instead depend on the type and properties of other components. For example, the positions of cylinder endpieces are determined by the presence and length of the cylinder halves. If the cylinder length is misclassified, the endpiece region will also be misaligned, potentially leading to incorrect classification. As a result, these regions require additional padding and targeted geometric augmentation during training.

4.1.2 Learning Tasks

Each region is associated with at least one specific learning task, depending on the properties of the components it may contain:

Presence Detection / Binary Classification Determines whether a region contains a specific component or is empty. This is used if only the presence or absence of an item needs to be predicted, without distinguishing between multiple variants.

Multi-Class Classification Applied if a region may contain one of several mutually exclusive component types. This requires categorical classification to determine which specific component, if any, is present.

Multi-Label Classification Used if a region can contain multiple components simultaneously. As categorical classification is inadequate, this task requires multi-class, multi-label classification or object detection techniques to identify the components.

Regression Involves predicting a numerical value associated with a region. In this study, regression is used to estimate the length of cylinder halves, the length of

Region	Positional Constraint	Learning Task
Red Tray	Tightly	Multi-Class
Electronic Thumbturns (x2)	Tightly	Multi-Class & Regression
Order Notes	Lightly	Multi-Label
Batteries (x4)	Tightly	Presence
Cam Lock Cylinder	Tightly	Presence
Euro Cylinder Cam	Tightly	Multi-Class
Euro Cylinder Halves (x2)	Tightly	Presence & Regression
Euro Cylinder Half Surface Finish (x2)	Relative	Multi-Class
Euro Cylinder Endpiece (x2)	Relative	Multi-Label
Swiss Cylinder Cam	Tightly	Multi-Class
Swiss Cylinder Halves (x2)	Tightly	Presence & Regression
Swiss Cylinder Half Surface Finish (x2)	Relative	Multi-Class
Swiss Cylinder Endpiece (x2)	Relative	Multi-Label
Scandinavian Cylinder	Tightly	Multi-Label
Padlock	Tightly	Presence

Table 4.1: Types of positional constraints and learning tasks for each region.

electronic thumbturn extensions, as well as the number of screws on the electronic thumbturns.

4.2 Datasets

Based on the regions listed in Table 4.1, we construct a total of 18 distinct datasets. Each dataset generally corresponds to a unique region of the workpiece carrier. Regions containing identical or symmetric components, such as thumbturns or batteries, are grouped into a shared dataset. The cylinder endpiece region is considered as four distinct datasets corresponding to the general type, dust cap presence, extension presence, and extension length. This allows each model to focus on a narrowly defined objective and its relevant inter-class distinctions, resulting in superior performance (see Section A.1).

Each dataset consists of the extracted image regions from the base dataset. While we generally retain the original image resolution, particularly large or small regions are adjusted to manage computational demands and to ensure appropriate feature scaling:

- Large regions are downscaled to reduce computational requirements.
- Small regions with fine details are upscaled to maintain sufficient resolution in the feature maps.

- Regions with large aspect ratios are adjusted to avoid inefficient padding or loss of information.

During training, each dataset is handled independently. However, the detection process for cylinders is a multi-step pipeline, as depicted in Figure 4.2. First, the cam and both halves of the large cylinders are cropped and classified individually. If a cylinder half is present, its length is estimated. This length is then used to determine the precise crop regions for the cylinder's endpiece and surface finish, which are subsequently processed. Models for the thumbturn extension and the dust cap are only applied if an electronic thumbturn attachment point was detected. If an extension is detected then a separate regression model is used to determine its length.

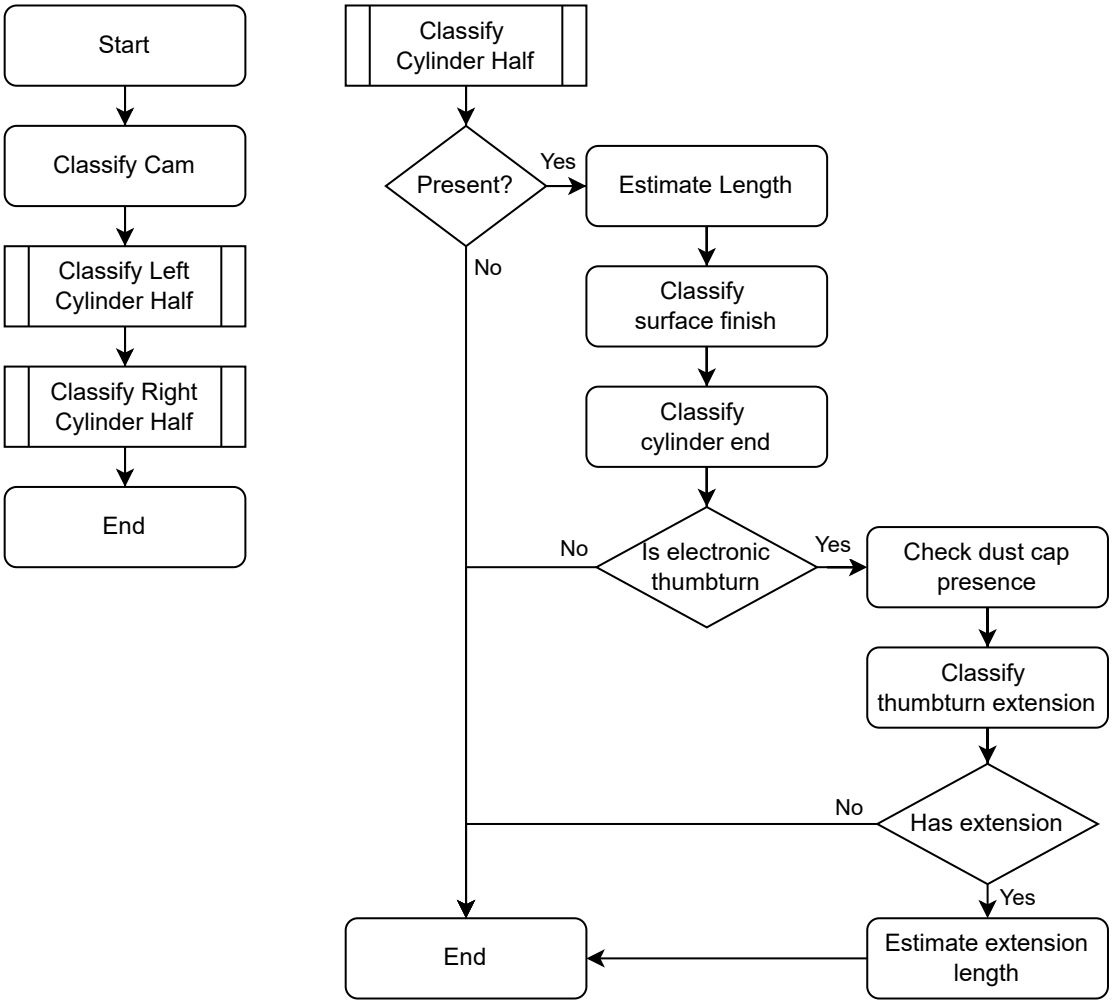


Figure 4.2: Diagram showing the steps to fully classify a Euro profile or Swiss round profile cylinder.

4.3 Augmentation

Data augmentation is critical for improving the robustness and generalization of our models. We apply distinct augmentation strategies for classification and regression tasks.

4.3.1 Classification

The classification datasets are augmented using AutoAugment [Cub+19], which applies one of several learned augmentation policies randomly to each image in a batch. We use the configuration trained on ImageNet, combined with CutOut [DT17] and horizontal or vertical flipping, depending on the symmetry of the region. Figure 4.3 shows a few examples of these transformations.

Figure 4.3: Example images demonstrating the classification augmentation pipeline, with the corresponding original images on the left.

4.3.2 Regression

For regression tasks, a modified RandAugment-based pipeline [Cub+20] is used. Geometric transformations such as rotation, scaling, and translation are disabled to avoid distorting the numerical relationships being predicted.

To address the highly imbalanced distribution of cylinder lengths, we additionally employ a custom horizontal shift augmentation. This involves randomly moving the cropped cylinder half images horizontally with a standard deviation of 30 pixels. The vacated edge region is filled with repeated border pixels. The effects of this augmentation are visualized in Figure 4.4.

Figure 4.4: Augmented cylinder halves for length estimation, shifted in both directions.

To assess the effect of this augmentation on data distribution, we visualize the resulting cylinder length frequencies in Figure 4.5. The augmented dataset size was made equal to the original to isolate the impact of the pixel-shift augmentation.

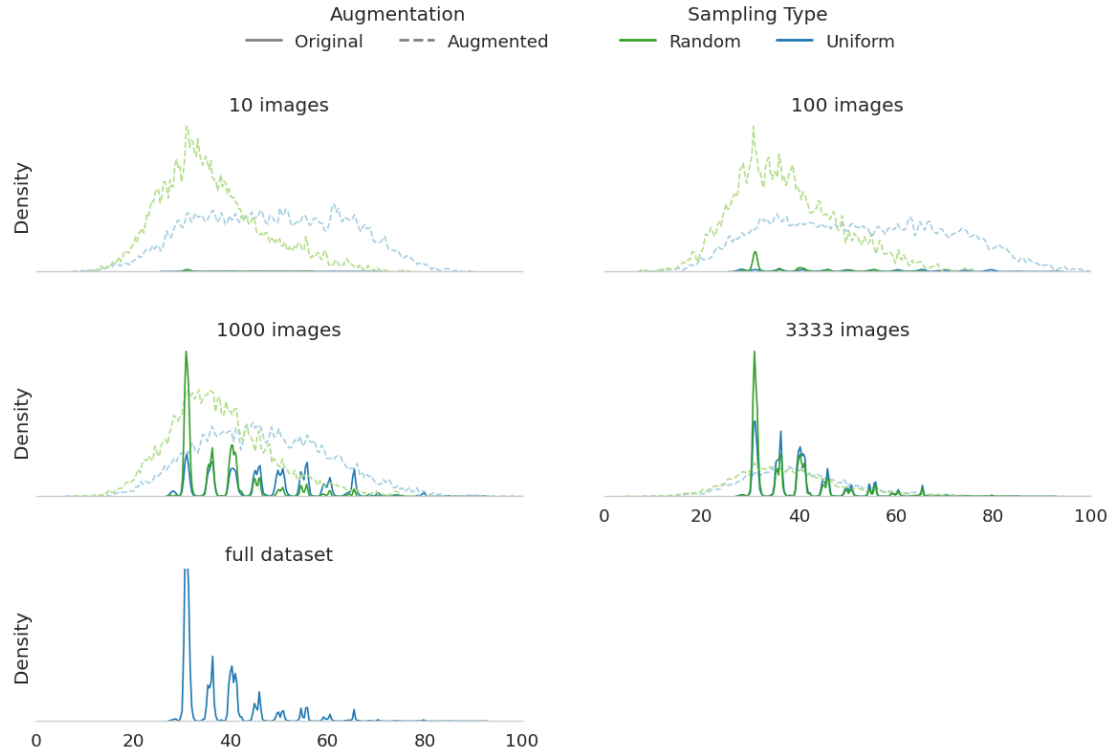


Figure 4.5: Distribution of cylinder lengths in the subsampled datasets before and after augmentation.

4.4 Model Architecture

We employ a modified ResNeXt-50 [Xie+17] architecture as our backbone, featuring a Global Average Pooling (GAP) neck and a custom classification head. As illustrated in Figure 4.6, the classification head consists of a Dropout layer (50% drop probability), followed by an intermediate fully connected layer with 512 units and ReLU activation,

and concludes with a variable-width output layer tailored to each task. The choice of ResNeXt-50 is motivated by its balance of efficiency and performance. It extends the widely used ResNet-50 by introducing grouped convolutions, which improve representational power without significantly increasing computational cost (see Figure 4.7). While other architectures may achieve superior performance on large-scale benchmarks like ILSVRC [Rus+15], our experiments show that this modified ResNeXt-50 consistently delivers the best performance on our specific dataset (see Section A.3.1).

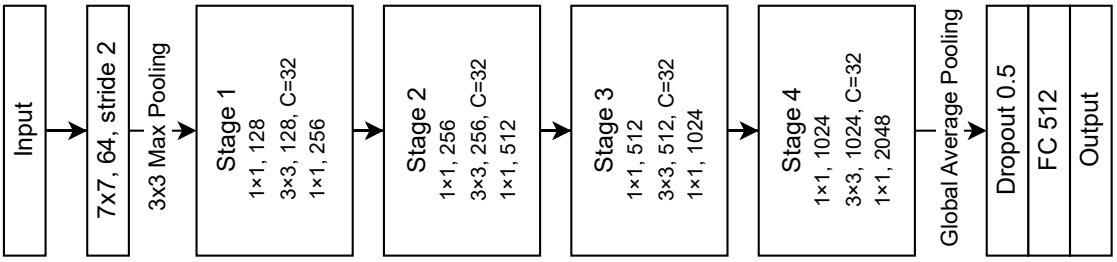


Figure 4.6: Schematic overview of the ResNeXt-50 backbone along with a custom classification head. Each stage corresponds to the block structure shown in Figure 4.7.

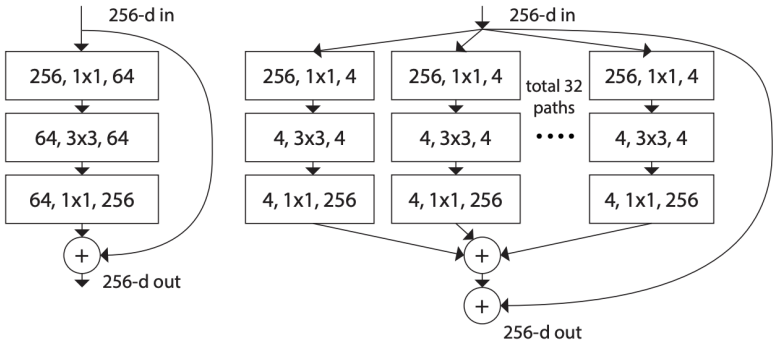


Figure 4.7: A block of ResNet (left) and a block of ResNeXt (right) of similar complexity from the work of Xie et al. [Xie+17].

4.5 Loss Functions

The choice of loss function depends on the learning task at hand:

- **Binary and Multi-Class Classification**: We use label-smoothed cross-entropy loss [Sze+16], which improves generalization by preventing the model from becoming overly confident in its predictions. Instead of assigning 100% probability to the correct class, a small fraction ϵ is distributed uniformly across all classes:

$$(1-\epsilon)\delta_{i,j} + \frac{\epsilon}{K}, \qquad \delta_{i,j} = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases} \tag{4.1}$$

where ϵ is the smoothing parameter, K is the number of classes, and $\delta_{i,j}$ is the Kronecker delta.

- **Multi-Label Classification**: We apply a modified version of label-smoothed cross-entropy that supports multiple correct labels per instance.
- **Regression**: For continuous outputs, we employ the Smooth L1 Loss [Gir15]. This loss function robustly handles outliers by behaving like the L2 (squared) loss for small errors and like the L1 (absolute) loss for larger errors, ensuring sensitivity to minor deviations while preventing exploding gradients.

$$\text{SmoothL1}(x) = \begin{cases} 0.5x^2, & \text{if } |x| < 1 \\ |x| - 0.5, & \text{otherwise} \end{cases} \tag{4.2}$$

4.6 Pretraining Strategy

We apply transfer learning with weights pretrained on ImageNet-1K [Rus+15]. Given the significant domain shift between ImageNet and the target datasets, we anticipate that most high-level features will have limited utility. Preliminary experiments, shown in Figure 4.8 and Figure 4.9, support this hypothesis and indicate that freezing the entire backbone or selectively reducing the learning rate results in either decreased performance or slower convergence. Consequently, we choose to freeze only the weights of the initial large convolutional layer, while using the full learning rate for the rest of the backbone.

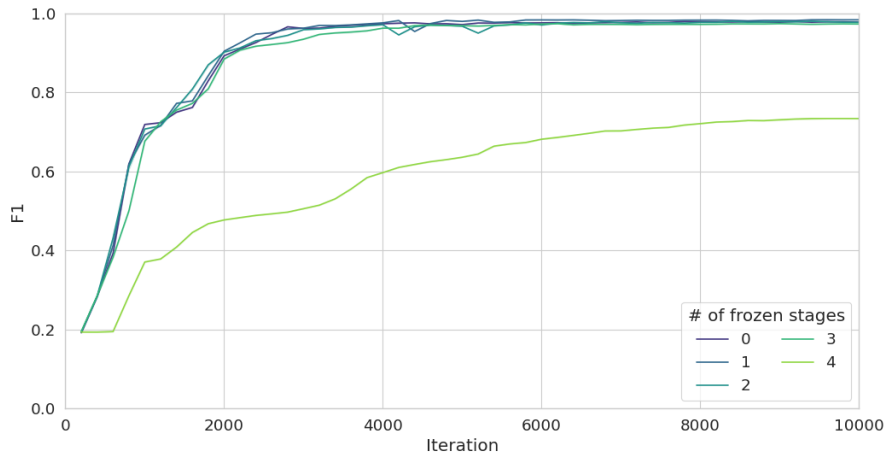


Figure 4.8: Performance comparison between fully retrained and partially frozen models.

4.7 Imbalance Mitigation

Many ROI-based datasets exhibit significant class imbalance, primarily due to two factors: unequal product popularity and the inclusion of an “empty” class, which can make up a

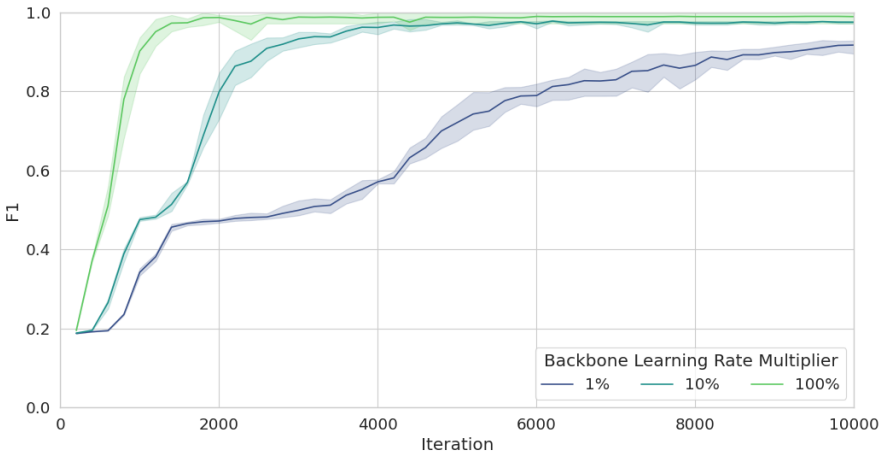


Figure 4.9: Performance comparison using different learning rate multipliers for the backbone.

significant proportion of the data for rarely populated regions. The degree of imbalance can be quantified using the imbalance factor (IF), computed as:

$$\mathrm{IF} = -\frac{\sum_{i=1}^{K} n_i \log_2(n_i)}{\log_2(K)} \tag{4.3}$$

where n_i is the number of samples in class i, n is the total number of samples, and K is the total number of classes.

The imbalance factor alongside dataset sizes across all regions is shown in Figure 4.10. Several datasets display extreme imbalance, requiring mitigation strategies to preserve model performance. We consider the following techniques:

- **Class Weighting** [KZ01]: Adjusts the loss function to emphasize minority classes and reduce the impact of majority classes. The weights are computed as:

  $$w_i = \min\left(\frac{n}{K \cdot n_i}, 10\right) \tag{4.4}$$

  to avoid excessively large weights for extremely rare classes.

- **Dataset Oversampling** [GDG19]: Duplicates minority class samples based on a repeat factor:

  $$\max\left(1, \sqrt{\frac{t}{f_c}}\right), \qquad f_i = \frac{n_i}{n} \tag{4.5}$$

  where t is an oversampling threshold and f_i is the class frequency.

- **Balanced Batch Sampling** [SLH16]: Adapts the sampling process to ensure that each batch contains a balanced combination of all classes.

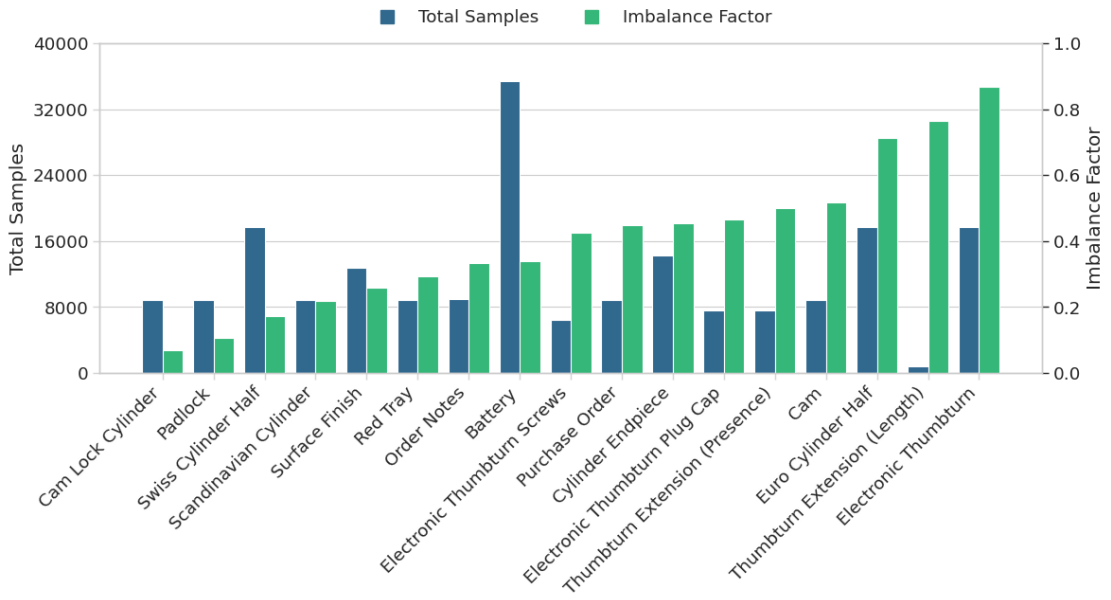


Figure 4.10: Imbalance factor and dataset sizes of all ROI-based datasets. A lower imbalance factor signifies greater imbalance.

- **Undersampling**: Reduces the influence of dominant classes by limiting each class to a maximum of 100 samples.

These methods are evaluated using a subset of region datasets with varying imbalance levels, as shown in Figure 4.11. The plot reports the average F1-score averaged across all training epochs, which reflects not only final accuracy but also convergence stability and consistency.

Among all tested methods, balanced batch sampling consistently yields the best performance across all tested datasets. Undersampling ranks second, highlighting the low intra-class variance that enables representative training even with fewer examples. The thumbturn extension dataset, which exhibits higher visual variability, benefits less from undersampling. In contrast, oversampling can achieve similar performance to subsampling if the highest thresholds are used, though lower thresholds slow convergence. Class weighting produces the most inconsistent results, even if class weights are clipped. Considering these findings, balanced batch sampling was selected for all subsequent experiments.

4.8 Training Setup

We use the AdamW optimizer, selected for its adaptability and robust handling of sparse gradients. The specific hyperparameter configuration for the optimizer and other settings can be found in Table 4.2. To further mitigate overfitting, we set weight decay to a relatively high value of 0.1.

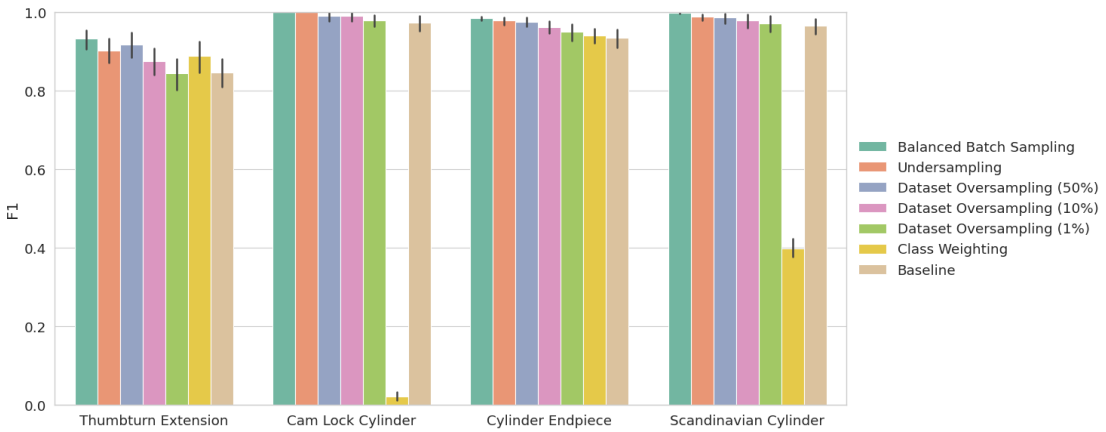


Figure 4.11: Comparison of imbalance mitigation strategies using average F1-score over training period.

Parameter	**Value**
Iteration	10,000
Batch size	64
Optimizer	AdamW
Base Learning Rate	0.001
Weight Decay	0.1

Table 4.2: Hyperparameter configuration used during model training.

We employ a two-phase learning rate schedule: an initial linear warm-up phase gradually increases the learning rate, followed by a cosine annealing schedule that reduces it over time. This strategy encourages stable convergence and helps the model escape early plateaus. The learning rate profile is illustrated in Figure 4.12.

4.9 Summary

This chapter presented ROI-based methods for automated assembly verification, leveraging predefined regions of interest on the workpiece carrier to transform object detection into a set of simpler classification and regression problems. By exploiting the spatial constraints of components, this approach enables the use of smaller models trained with only image-level labels, significantly reducing annotation effort compared to conventional object detection.

Regions were categorized based on spatial constraints and associated learning tasks, enabling tailored training and models for each. We introduced our custom augmentation strategies and a modified ResNeXt-50 architecture. Among various approaches, balanced batch sampling proved most effective in mitigating class imbalance, thereby maintaining

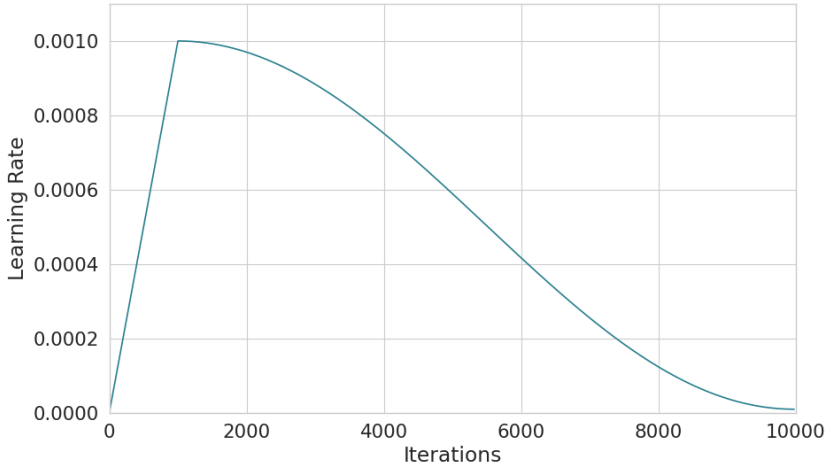


Figure 4.12: The learning rate schedule used for training: linear warmup followed by a cosine annealing schedule.

consistent model performance across diverse datasets.

CHAPTER 5

Object Detection

This chapter explores object detection as an alternative to the region-of-interest (ROI) methods discussed in the previous chapter. Object detection does not rely on constrained object locations, making it more robust to misplacements and variations in positioning. It also simplifies the process by using a single model instead of one per region. Dedicated regression models also become mostly unnecessary, as the length of the cylinder halves and extensions can be directly derived from the predicted bounding boxes.

However, certain challenges arise with object detection. First, it requires precise bounding box annotations, which are about 10 to 20 times more time-consuming to create than image-level labels. While object detection can theoretically identify misplaced or rotated objects, the scarcity of such examples in the dataset means it is far from guaranteed that these cases will be detected reliably. Additionally, since object detection works at the carrier level, it limits the ability to resample minority classes or fine-tune models for specific components. Lastly, some properties, such as the surface finish, are not straightforward to detect with object detection.

This chapter focuses on optimizing object detection for lock assembly verification. We explore both standard object detection methods as well as supplemental strategies involving synthetic data and pseudo bounding boxes. These methods are designed to improve data efficiency, enhance model resilience, and reduce annotation costs.

5.1 Pseudo-Annotation

Although object detection does not require fixed object positions, the constrained layout enables the generation of approximate bounding boxes automatically from image-level labels. However, this method only provides accurate annotations for correctly positioned components, limiting its effectiveness for parts with variable size, orientation, or placement.

These pseudo bounding boxes are initially derived from the carrier regions used in the ROI-based approach and then refined based on typical component dimensions and relative positioning. Examples of these generated annotations are shown in Figure 5.1.

(a) Image with inaccurate annotations for the order note and a missing annotation for the dust cap.

(b) Image with inaccurate order note but correct dust cap and thumbturn extension annotations.

Figure 5.1: Two example images with pseudo-annotations.

The distribution of IoU values between the pseudo bounding boxes and their corresponding ground truth boxes is illustrated overall in Figure 5.2 and per class in Figure 5.3. While the pseudo bounding boxes generally provide a reliable approximation, some component classes exhibit significantly lower overlap. The order notes placed in the large flat area for extra parts exhibit greater positional variance, resulting in pseudo bounding boxes that are often imprecise or inaccurate. Similarly, the yellow purchase orders face comparable issues due to variability in their extents. In such cases, manual labelling could be used to improve performance if required (see Table 7.1).

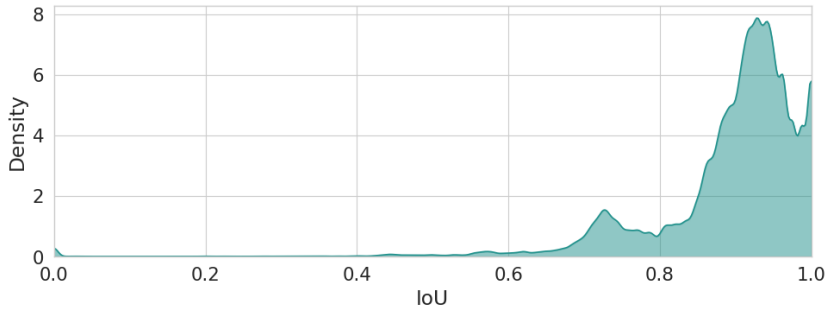


Figure 5.2: Distribution of IoU values between corresponding pseudo and ground truth annotations across all classes.

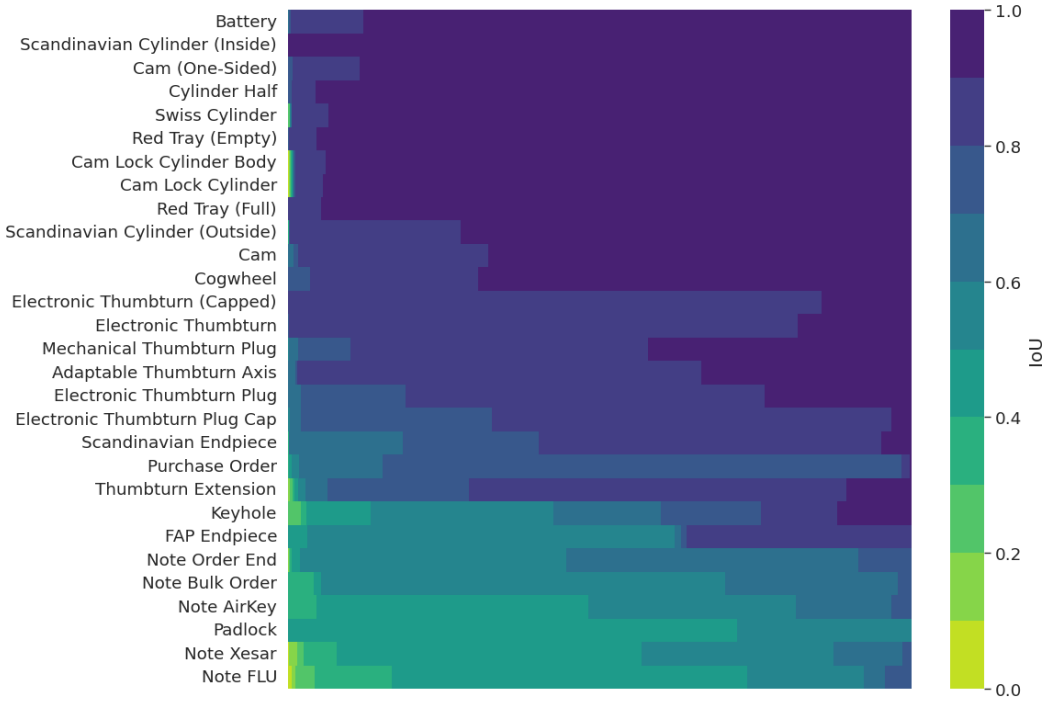


Figure 5.3: Distribution of IoU values between corresponding pseudo and ground truth annotation per class.

5.2 Synthetic Carrier Generation

The primary advantage of object detection over ROI-based methods is its ability to handle misplaced or rotated parts. However, this strength is limited by the scarcity of such cases in the training data. Combined with the overall imbalance and low diversity of the dataset, this makes the model susceptible to overfitting. To mitigate these issues, we explore synthetic data generation through compositional recombination to create a more balanced and varied training set.

The synthetic generation process follows the steps outlined in Figure 5.4. We begin by extracting individual object annotations from the dataset, creating a collection of individual parts for later use. For each generated image, a background is chosen, consisting of either a nearly empty image from the source dataset (50%) or a random image from the COCO dataset [Lin+14] (50%). Two cylinders and up to five randomly chosen parts are then generated and randomly placed on the background with a chance for a small rotation of up to 15°, ensuring that overlap between parts does not exceed 20%.

For the cylinders, we first randomly select a cylinder type, roughly weighted based on their relative variability. Thus, for each required part a random image is chosen from the part collection and positioned according to predefined offsets. To avoid hard outlines and transitions, each part image includes a small margin of surrounding pixels. This margin

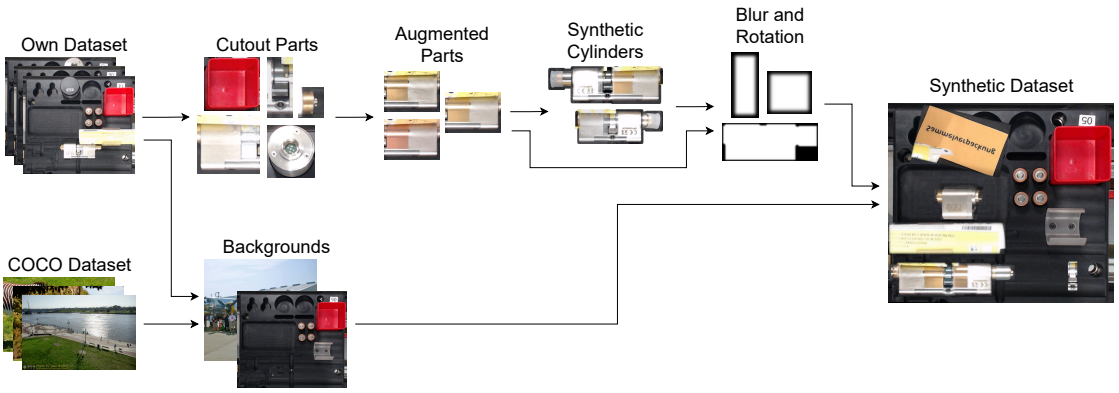
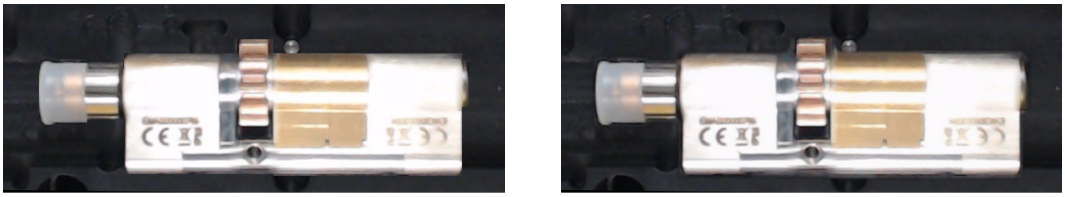


Figure 5.4: Overview diagram of the procedure to generate synthetic workpiece carriers.

is used to blend together neighbouring parts and reduce the artificial appearance of the generated cylinder. The outer contour is similarly gradually faded to transparency over a 10-pixel width.



(a) Hard Outline (b) Blended outline

Figure 5.5: Comparison of hard and blended outlines of generated lock cylinders.

(a) Hard transition (b) Blended

Figure 5.6: Comparison of outline and part boundary of synthetic cylinders before and after treatment.

To introduce further diversity, parts are reused across cylinder types and mirrored to fit either side when applicable. Additionally, all parts undergo augmentation, including brightness, contrast, and sharpness adjustments, as well as random vertical flipping, when appropriate. Finally, in addition to the cylinders, five random parts are placed within the image to help the model learn individual components in different contexts.

The selection process for the cylinder and its individual parts is guided by dataset balance. Each time a part is chosen, its type is probabilistically determined based on the current

class imbalance, considering both the base dataset annotations and the parts already generated. Once the type is selected, a specific part is randomly selected from the available options within that category.

Figure 5.7 presents several examples of generated images created using this procedure.

Figure 5.7: Examples of synthetic images generated using the described data generation pipeline.

Our testing has shown that relying solely on generated data is not optimal. For this use case, we found that combining the synthetic dataset with the original one in a 2:1 ratio yields the best results.

5.3 Model Architecture

We use a Faster R-CNN [Ren+17] for its strong performance across a wide range of applications and its ability to generate accurate bounding boxes. Since our application

does not have strict computational constraints, we can take advantage of Faster R-CNN's accurate detections, which stem in part from its two-stage design.

Transformer-based models were not extensively considered due to the limited availability of training data, which impacts them more than CNN based architectures. Similarly, single-stage models were not investigated intensively, as this use case does not impose strict computational constraints. For feature extraction, we selected the ResNet-50 backbone. Although ResNeXt-50 yielded the best results in our ROI-based methods, ResNet-50 slightly outperforms it in the object detection experiments. This difference is likely due to the availability of pretrained weights specifically optimized for object detection with ResNet-50, whereas only image classification pretrained weights are available for ResNeXt-50. After the feature extractor, we apply Dropout with a drop probability of 20%.

To further enhance performance, we integrate two techniques: Generalized IoU (GIoU) Loss [Rez+19] and Online Hard Example Mining (OHEM) [SGG16]. GIoU Loss improves bounding box regression by addressing a key limitation of traditional IoU loss, which produces zero gradients if there is no overlap between predicted and ground truth boxes. By considering the area of the smallest box that can enclose both predictions, GIoU always gives a meaningful gradient, allowing the model to steadily improve its localization even with slight misalignments. This is crucial for our dataset, where accurate detection of small parts is required. On the other hand, OHEM prioritizes difficult samples during training by focusing on those with high loss. This helps the model learn from rare cases that cannot be easily oversampled, because they do not form distinct classes, such as misplaced objects.

5.4 Augmentation

To increase the effective dataset size, we use the augmentation pipeline depicted in Figure 5.8, which is based on the one used for RTMDet [Lyu+22], which itself is based on the one used for YOLOX [Ge+21]. This pipeline incorporates resizing, cropping, color jittering, and flipping. For most of the training, except for the last epochs, the pipeline also includes stronger augmentation: Mosaic (Figure 5.9a) and MixUp [Zha+18] (Figure 5.9b). In contrast to the original RTMDet pipeline, we introduce several modifications to further improve augmentation. Specifically, we adapt the resize and crop sizes, apply random stretching, and increase the resolution of intermediate images to provide more detailed features for the model to learn. Some examples of augmented images can be seen in Figure 5.10. Unlike for ROI-based classification, we refrain from using AutoAugment or similar here, as it includes rotations and shears, which result in unrepresentative bounding boxes and degraded performance.

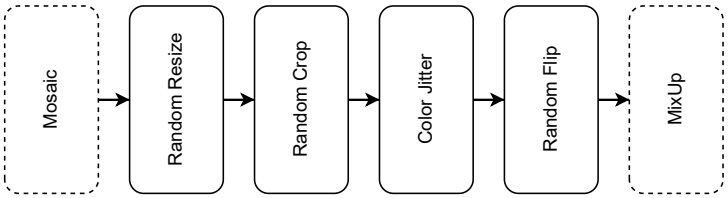


Figure 5.8: Overview of the data augmentation pipeline with steps active throughout training with solid strokes and those disabled during the final epochs with dashed strokes.

(a) Mosaic (b) MixUp

Figure 5.9: The Mosaic and MixUp steps of the data augmentation pipeline applied to two example images.

Figure 5.10: The full data augmentation pipeline applied to some example images.

5.5 Summary

This chapter presents object detection as a flexible alternative to previous region-of-interest (ROI) methods for lock assembly verification. Object detection does not require fixed object positions and simplifies processing by using a single model and removes the need for separate regression models, as object sizes can be inferred directly from predicted bounding boxes. To mitigate the high effort required for dataset annotation we proposed two supplemental strategies with pseudo annotation and synthetic data generation.

The chapter introduces the model we employ, a Faster R-CNN with a ResNet-50 backbone, enhanced by GIoU Loss and OOHEM. For data augmentation, an adapted pipeline inspired by RTMDet and YOLOX is used, including Mosaic and MixUp techniques. Together, these strategies form a strong baseline, which we use to evaluate our data-centric methods.

CHAPTER 6

Evaluation

This chapter outlines the experimental setup used to assess model performance. It includes details on dataset preparation, cross-validation strategy, construction of subsampled datasets, and the metrics employed for classification and detection tasks.

6.1 Dataset Partitioning

To ensure reliable evaluation, the collected dataset is first divided into a training/validation set, and a held-out test set, using an 80:20 split. This split is performed using stratified grouped sampling, which ensures the class distribution in each subset mirrors that of the original dataset [Coc77]. Using random sampling instead could introduce bias or lead to unrepresentative evaluation results [FS10; Koh95]. Since each image generally contains multiple annotations, disjoint partitioning is required. To further ensure testing accuracy, extremely rare classes are over-represented in the test split with a minimum of 10 samples per class. The resulting dataset distribution of each split can be seen in Figure 6.1.

The larger portion of the dataset allocated for training and validation is further divided using stratified 3-fold cross-validation. This technique splits the data into three distinct folds, where each fold is used as a validation set once while the remaining folds are combined for training [KV94]. A value of $k = 3$ is selected to strike a balance between a manageable number of training runs and a sufficiently sized validation set for rare classes. Stratified cross-validation maintains the class distribution across folds, which is crucial for handling rare classes that might otherwise be distributed very unequally. Although small validation sets can lead to variability due to dataset shift [Qui+22], averaging results over multiple folds mitigates this issue and leads to more reliable performance estimates without reducing the training set size [FS10; KV94].

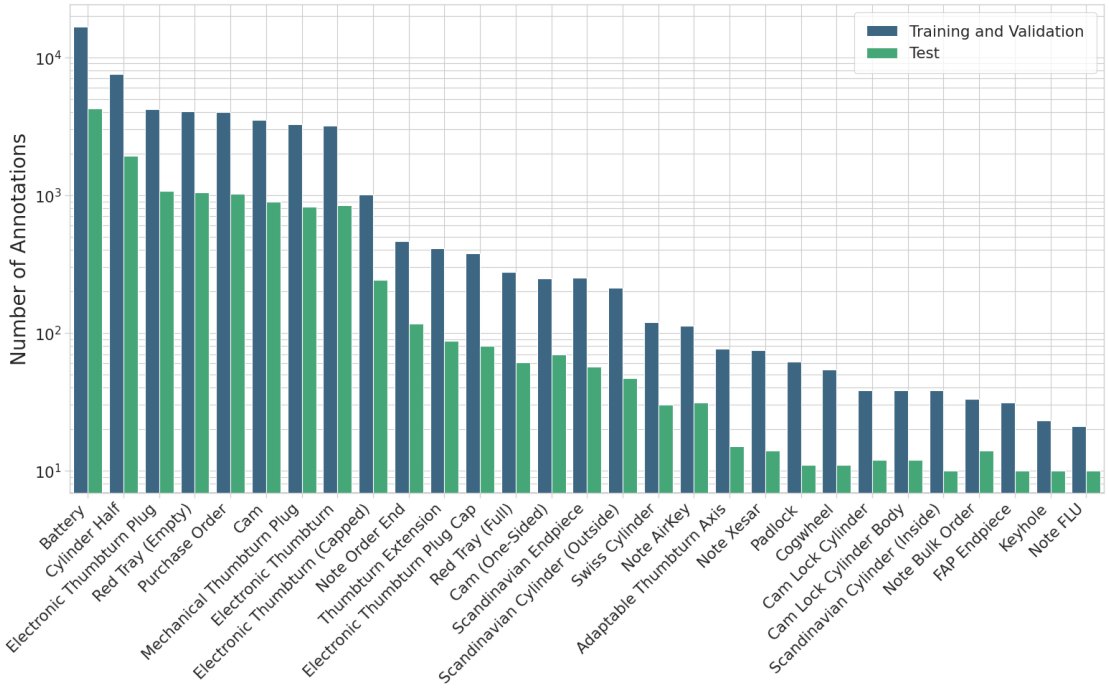


Figure 6.1: Number of annotations for each category in the train/validation and test split (logarithmic).

6.2 Dataset Subsampling

Since the focus of this study is on the performance characteristics of small datasets, we also consider subsampled datasets of varying sizes based on the training set of each cross-validation fold. This setup also allows us to compare two types of datasets: large, automatically collected datasets that are often noisy and imbalanced, and smaller, manually curated datasets that are usually of higher quality and more balanced. Subsampling enables a controlled analysis of how the amount and composition of training data affects performance.

Subsampling is constrained by the class distribution of the original dataset. Since some categories are rare, it is not always possible to sample an equal number of examples for every class. As a result, smaller subsets tend to be more balanced, while larger ones reflect the original imbalance more closely. Since training is iteration-based, the dataset size does not affect training duration, allowing for a fair comparison of the impact of dataset size.

Two types of subsampled datasets are used in this study, corresponding to the differing requirements of ROI-based methods and object detection tasks. Consequently, results cannot be directly compared between the two approaches, except for those in Section 7.3.

Minimum annotation count per category	Number of images
1	13
3	37
10	127
100	717
full	2951

Table 6.1: Number of images in the training split for each of the object detection subsampled datasets used during evaluation.

6.2.1 ROI-Based Methods

For ROI-based tasks, subsampled datasets are created by randomly selecting a fixed number of images per class from the training splits of each fold. From each selected image, regions of interest (ROIs) corresponding to the target class are extracted and used as individual training samples. This results in a class-balanced dataset, assuming sufficient samples of each class are available.

6.2.2 Object Detection

For object detection, each dataset is constructed by selecting images until each class reaches a predefined minimum number of annotations. Because images contain multiple instances, selection is performed iteratively by always choosing an image containing an instance of the most under-represented class. Thus, only the rarest classes are guaranteed to match the specified annotation minimum, while more common categories may end up over-represented. The average number of images in the resulting object detection datasets is shown in Table 6.1, and the corresponding class distributions are visualized in Figure 6.2.

6.3 Metrics

6.3.1 F1-Score

To evaluate the classification performance we mainly use the F1-score, a commonly used metric combining precision and recall into a single value. Precision is the fraction of correctly identified positive samples, while recall is the fraction of positive samples identified by the system. Given the inherent trade-off between these metrics, the F1-score strikes a balance, as it is calculated as the harmonic mean between them:

$$\mathrm{F1} = \frac{2}{\frac{1}{\mathrm{Precision}} + \frac{1}{\mathrm{Recall}}} \tag{6.1}$$

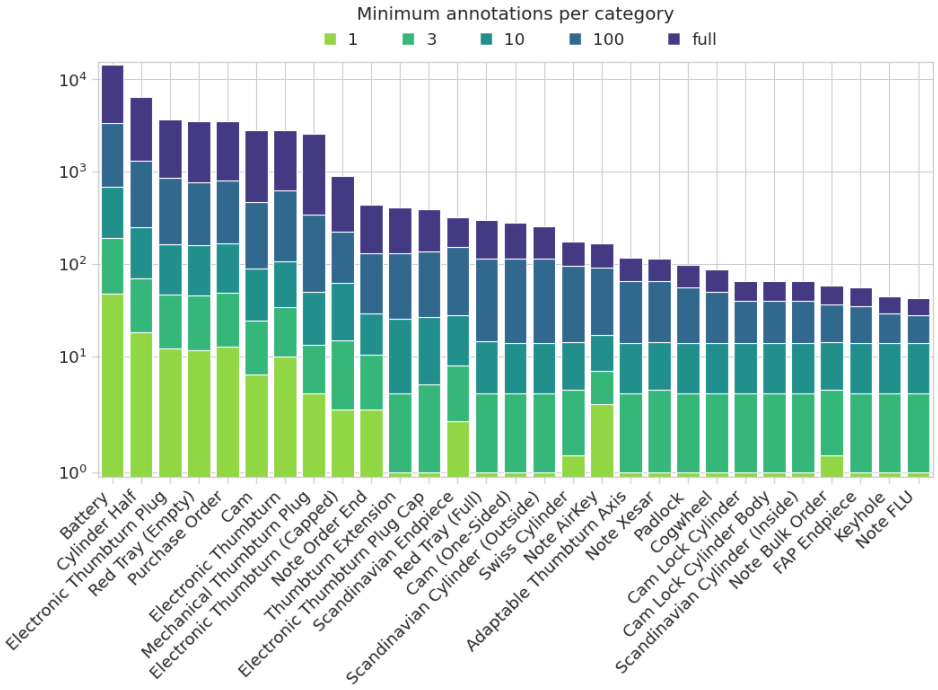


Figure 6.2: Distribution of annotation categories across the various object detection subsampled datasets used in the evaluation.

For multi-class classification, we use the macro F1-score, which averages the class-wise F1-scores. This approach is preferred for imbalanced datasets, as it ensures that performance is equally considered for all classes, rather than focusing on majority classes.

6.3.2 Balanced Root Mean Squared Error (BRMSE)

For regression tasks, we use the Root Mean Squared Error (RMSE) to measure the average difference between predicted and actual values. Since the ground truths of the cylinder and extension lengths, as well as the screw counts, are each from a distinct set of values, the output domain can be divided into corresponding bins. Since the datasets are highly imbalanced, it is important to avoid bias. As such, we compute the RMSE for each subrange and average the individual metrics to get the balanced RMSE.

$$\text{BRMSE} = \frac{1}{K}\sum_{k=1}^{K}\sqrt{\frac{1}{n_k}\sum_{i\in k}(y_i - \hat{y}_i)^2} \tag{6.2}$$

6.3.3 Average Precision (AP)

The Average Precision (AP) metric, as defined by the COCO benchmark, is the standard measure for object detection performance. It evaluates how well a model detects objects

by integrating the precision-recall (PR) curve. The PR curve plots recall on the x-axis and precision on the y-axis for different confidence thresholds. The standard COCO evaluation protocol computes the AP for 50% (AP_{50}) and 75% (AP_{75}) IoU thresholds, as well as the average AP from 50% to 95%, in 5% increments, referred to simply as AP.

CHAPTER 7

Results

This chapter presents the evaluation results for the proposed methods. We begin by assessing the baseline object detection approach, followed by the results of applying pseudo-annotations and synthetic data generation. Next, we evaluate the proposed ROI-based methods on classification and regression tasks. Finally, we consider the trade-off between dataset imbalance and size.

7.1 ROI-Based Methods

7.1.1 Classification

We begin by evaluating the performance of ROI-based classification methods. As shown in Figure 7.1, the models achieve consistently high F1-scores across nearly all classes when trained on the full datasets. Remarkably, even with substantial subsampling, the methods remain effective: with as few as three samples per class, F1-scores above 90% can be attained. Expanding to 10 samples per class typically captures the key variations within each category, leading to near-perfect performance. Beyond this point, additional data yields only marginal gains. Although the larger datasets used in the experiments may seem excessive in the context of data-efficient learning, they require little manual annotation effort, since most labels can be generated automatically during data acquisition.

There are two notable exceptions to these findings. First, the thumbturn screws require more images overall, likely because missing screws are less visually distinct than other appearance variations. Second, performance on the order notes seems to have a limit. This is likely due to a few variants that differ only in the handwritten text written on them, while sharing similar strokes and backgrounds, making them difficult to distinguish. Frequent occlusions and a low number of instances further contribute to this apparent limit.

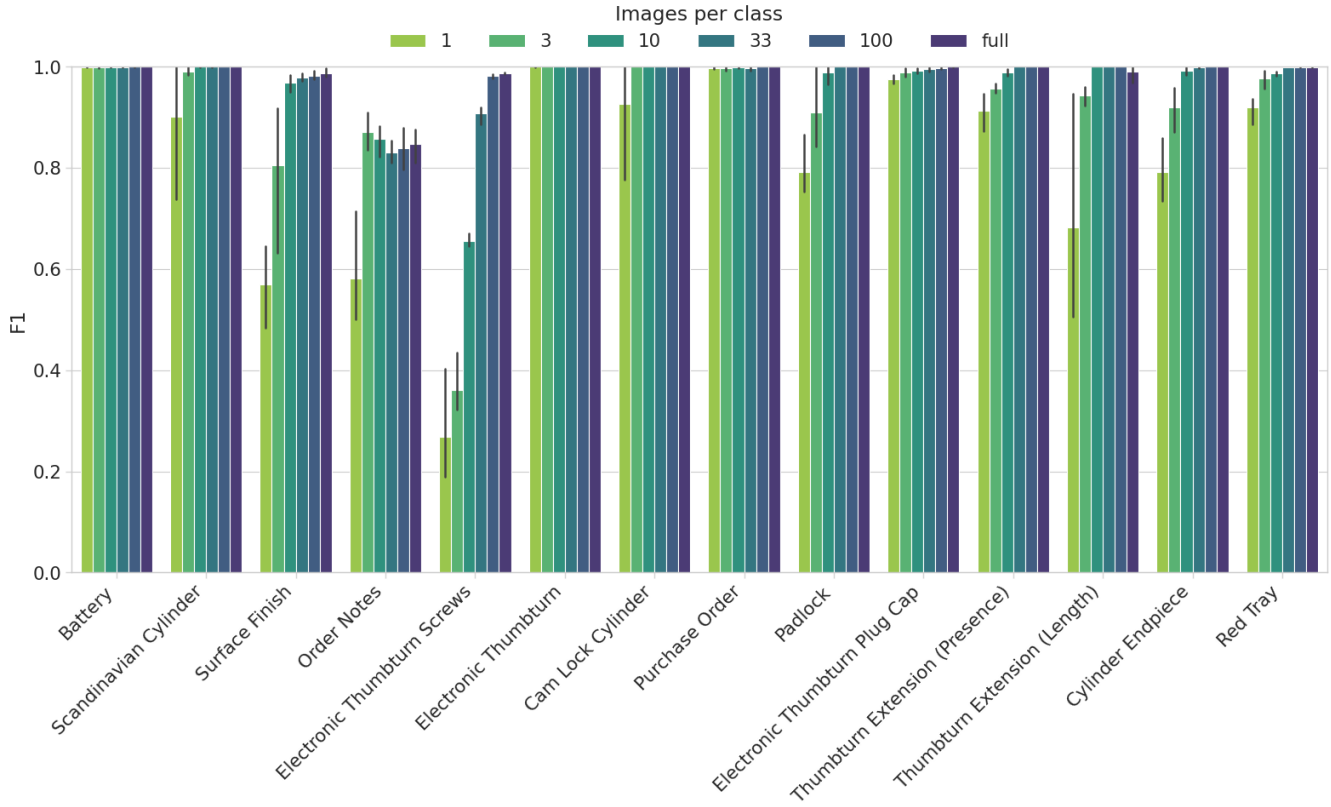


Figure 7.1: Results on the full datasets and subsampled variants based on top F1-score averaged over all folds.

7.1.2 Regression

We now transition from classification to regression models, which predict a continuous numerical output using a different loss function and augmentation pipeline.

Cylinder Length Estimation

We begin by estimating the lengths of the cylinder halves. To evaluate performance, we compare the proposed model trained on the full baseline dataset with versions trained on randomly and uniformly subsampled data. The results, along with the effect of the custom shift augmentation, are shown in Figure 7.2. The strong correlation between balanced accuracy and balanced RMSE indicates that the model is learning to estimate the actual lengths as continuous values, rather than simply selecting the correct bin. On large datasets, the model can achieve over 99% balanced accuracy and an BRMSE approaching the annotation margin of ± 1 mm. Even when trained with only 10 or 100 images, it maintains high accuracy of 94% and 98%, respectively, demonstrating robust performance with limited data.

The models trained on uniformly sampled datasets consistently outperform their randomly sampled counterpart when no augmentation is applied, underlining the importance of maintaining a balanced training distribution. With smaller datasets, augmentation is essential for acceptable performance, with uniformly sampled datasets clearly favoured. As dataset sizes increases, the effect of augmentation lessens. Augmented randomly sampled as well as unaugmented uniformly sampled datasets perform best by a slim margin.

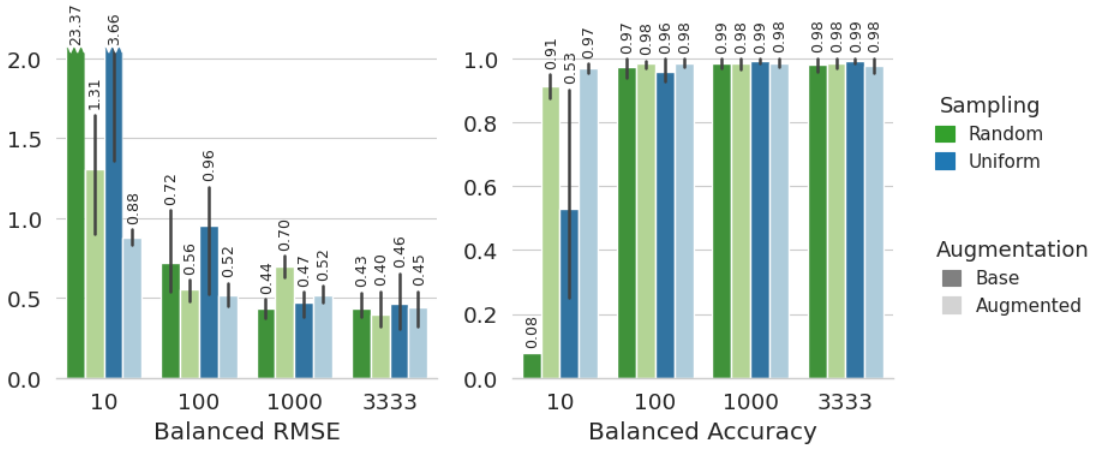


Figure 7.2: Cylinder length estimation results across various dataset sizes and augmentation strategies.

Other Regression Tasks

In addition to cylinder length, two further regression tasks are evaluated: screw counting on electronic thumbturns and estimation of thumbturn extension length. Both tasks involve only a small number of discrete values (three and four, respectively), and can therefore also be approached as classification problems. The corresponding classification results were already included earlier in Figure 7.1.

In Figure 7.3, we show a comparison of regression and classification performance on these tasks using differently sized datasets. For screw counting, regression yields superior results across most dataset sizes, while classification performs better for measuring the thumbturn extension length. Overall, classification tends to outperform regression on smaller datasets, while regression shows better results on larger datasets, particularly if the output domain is wide, as in the case of cylinder lengths, and sufficient training data is available.

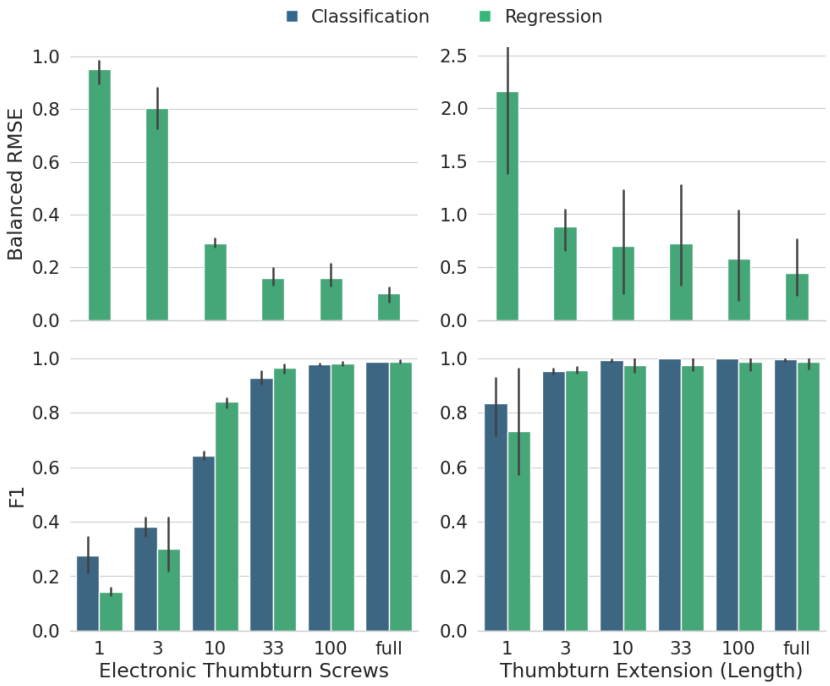


Figure 7.3: Comparison of regression and classification models for screw counting and thumbturn extension length estimation. Top: regression BRMSE; Bottom: F1-score comparison of both approaches.

7.2 Object Detection

In this section, we evaluate object detection as an alternative to ROI-based methods. Although object detection requires more detailed annotations, it offers greater flexibility and robustness, particularly in handling diverse layouts and configurations. We begin by assessing performance on a manually annotated baseline dataset, then extend the evaluation to include models trained on pseudo-labelled and synthetic datasets.

7.2.1 Baseline

We begin by presenting the results on the baseline dataset in Figure 7.4, it compares model performance across varying dataset sizes, showing both AP and AP_{50} scores for different minimum annotation thresholds per class. As expected, model performance improves as dataset size increases. However, moving from a minimum of 100 annotations per class to the full dataset does not yield further improvements and even shows some performance regressions. This is likely because this step primarily adds images containing only the most common classes and configurations.

For smaller datasets, performance is uneven across classes. Rare classes that barely meet the annotation threshold, such as uncommon order notes or less frequent cylinder end

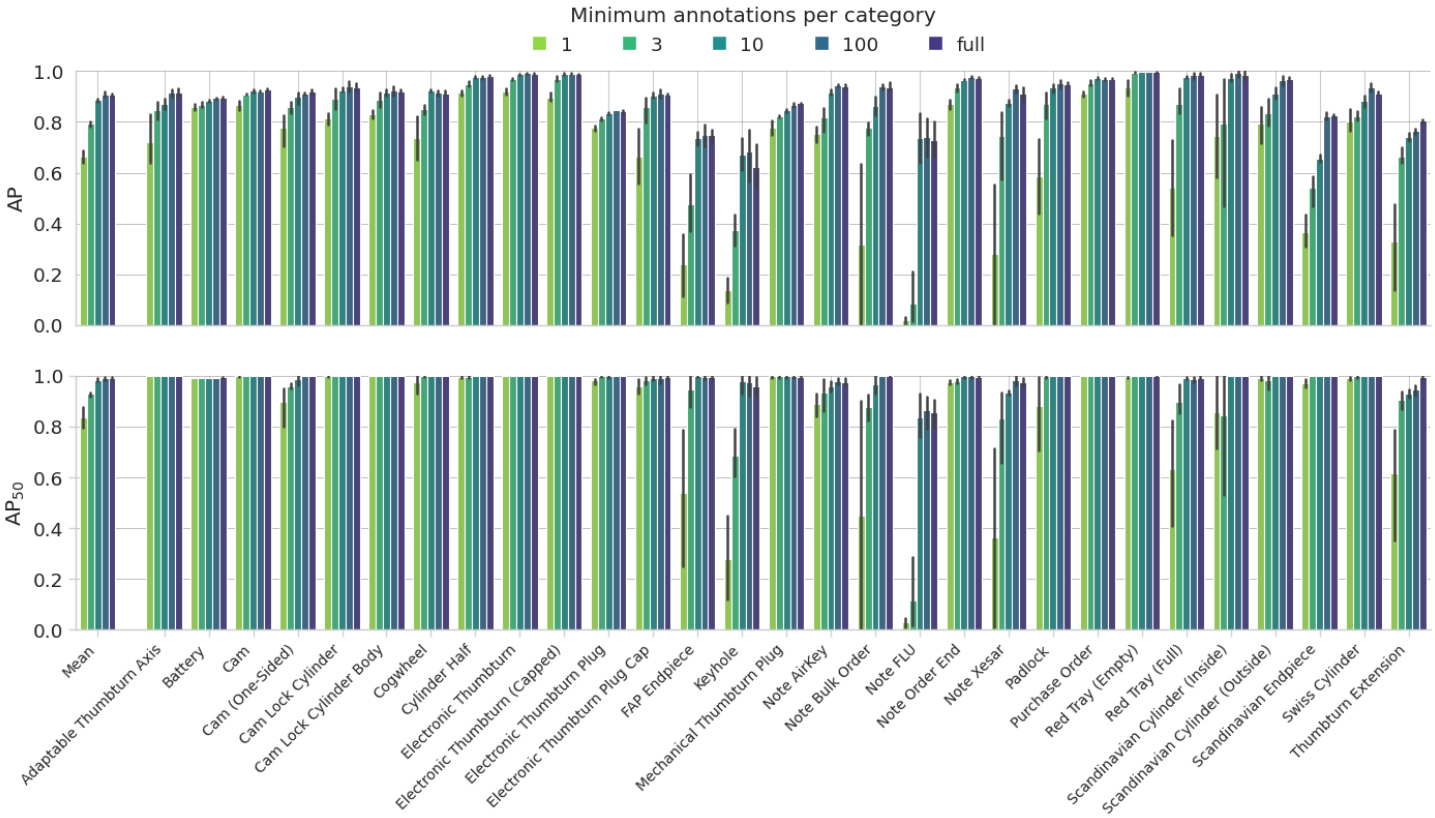


Figure 7.4: Baseline performance on various dataset sizes, measured by average precision (AP) and AP_{50} for different per-class annotation thresholds.

pieces, tend to show significantly lower AP than more common components. Since the intended application focuses on verifying component presence and type on the workpiece carrier, rather than precise localization, AP_{50} is the more relevant metric. Encouragingly, AP_{50} reaches near-perfect levels for most classes if as few as 10 annotated examples per class are included.

7.2.2 Pseudo-Annotation

Next, we evaluate our proposed approach based on pseudo bounding boxes. Figure 7.5 compares the performance of models trained solely on pseudo bounding boxes against those of baseline datasets of varying sizes. Due to the inherent imprecision of pseudo bounding boxes, the observed AP is lower than that of even the smallest baseline dataset. However, the approach remains effective at detecting and classifying the objects as reflected in the AP_{50} and especially the non-standard AP_{25} scores. Notably, the pseudo annotation method achieves an AP_{25} of 96.7%, approaching the performance of the datasets with at least 10 annotations per category, despite relying solely on weakly-supervised labels.

Figure 7.5: Comparison of the models trained on the baseline datasets compared to using only pseudo bounding boxes using AP, AP_{50}, and AP_{25} scores, demonstrating the high classification but low localization performance.

The poor localization performance stems primarily from a few select classes, as can be seen in Figure 7.6. Specifically, order notes located in the large, flat, loosely constrained general-purpose area have highly inaccurate pseudo ground truth annotations. Similarly, padlocks and, to a lesser extent, purchase orders (yellow paper) also suffer from inaccurate pseudo-annotations. As a result, the model can only replicate these imprecise annotations, leading to low IoU scores for these classes. Figure 7.7 highlights two examples of these problematic cases by comparing the ground truth, pseudo ground truth, and resulting predictions. While the model successfully aligns with the pseudo ground truth, it is unable to compensate for the inherent localization inaccuracies.

Since these issues are limited to a few classes, it is possible to supplement the pseudo-annotations with manual bounding boxes for these classes, which can improve the results of the pseudo dataset with minimal effort. Assuming the performance for the manually annotated classes matches those of the source dataset, we can estimate the impact on performance and efficiency of supplementing specific classes. Given that the largest subsampled dataset, with at least 100 annotations per category, achieves almost identical performance to the full dataset while using fewer images, we base our estimations on it. The improvements and effort involved in supplementing specific classes with manual annotations are thus shown in Table 7.1 and Figure 7.8. The data demonstrates that using only a few hundred manual annotations can elevate the AP_{50} to almost match the full dataset while also significantly improving the AP.

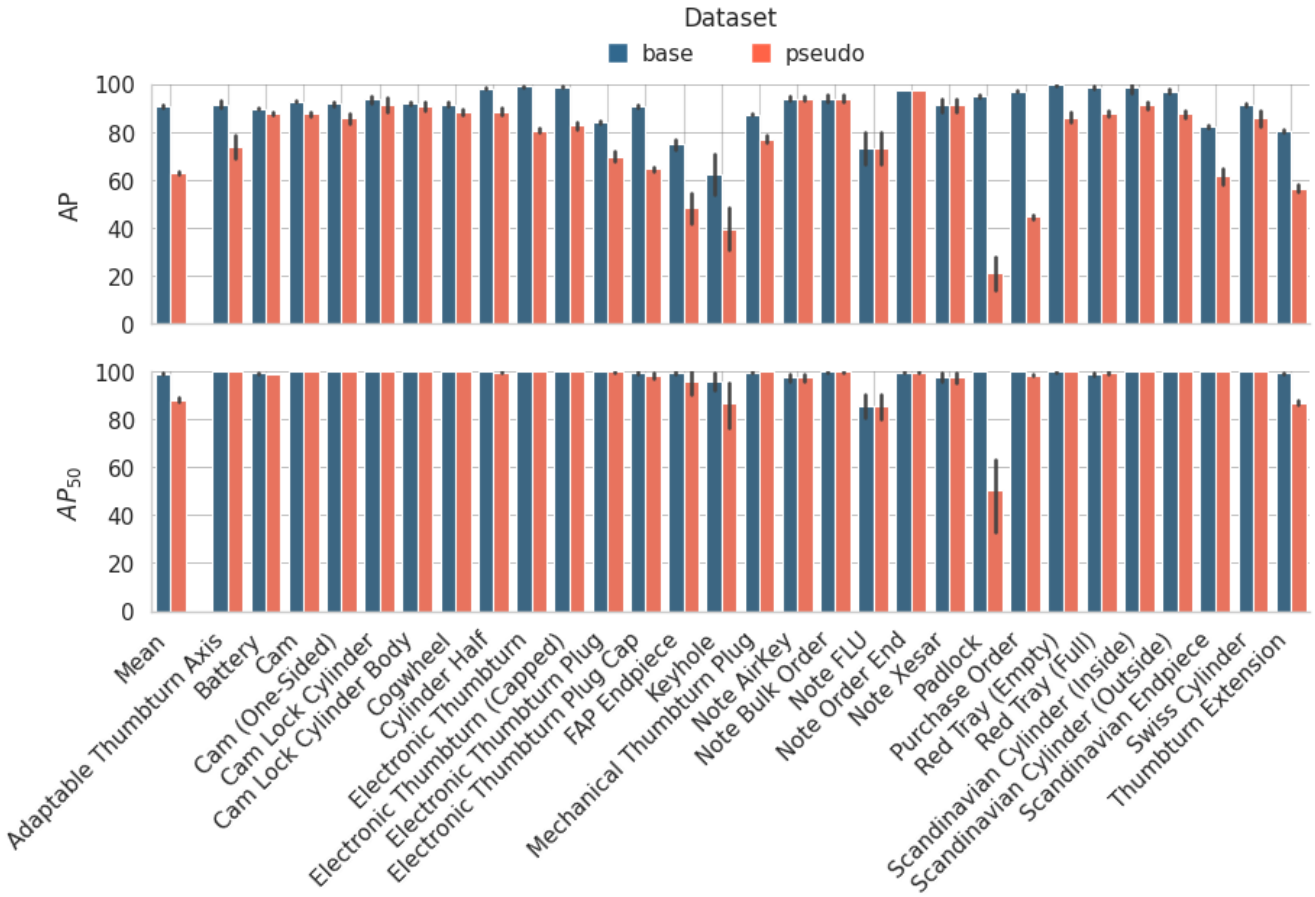


Figure 7.6: Comparison of class-wise results using the baseline and pseudo datasets.

Dataset	**Manual ann.**	**AP**	**AP_{50}**	**AP/100 ann.**	**AP_{50}/100 ann.**
Pseudo	0	63.06	88.28		
+ Padlock	41	+ 2.54	+ 1.70	+ 6.15	+ 4.12
+ Notes	261	+ 13.76	+ 8.10	+ 5.28	+ 3.11
+ Keyway	15	+ 0.79	+ 0.32	+ 5.18	+ 2.08
+ FAP Endpiece	21	+ 0.92	+ 0.13	+ 4.45	+ 0.61
+ Electronic Thumbturn Plug Cap	110	+ 0.91	+ 0.04	+ 0.83	+ 0.04
+ Thumbturn Extension	105	+ 0.83	+ 0.43	+ 0.79	+ 0.41
+ Electronic Thumbturn	520	+ 2.95	+ 0.03	+ 0.57	+ 0.00
Subsampled ($\geq$ 100)	8185	90.64	99.04		

Table 7.1: Impact of supplementing pseudo bounding boxes with manual annotations for specific classes. The table shows the increase in AP and AP_{50} scores, along with annotation efficiency, measured by AP per 100 annotations.

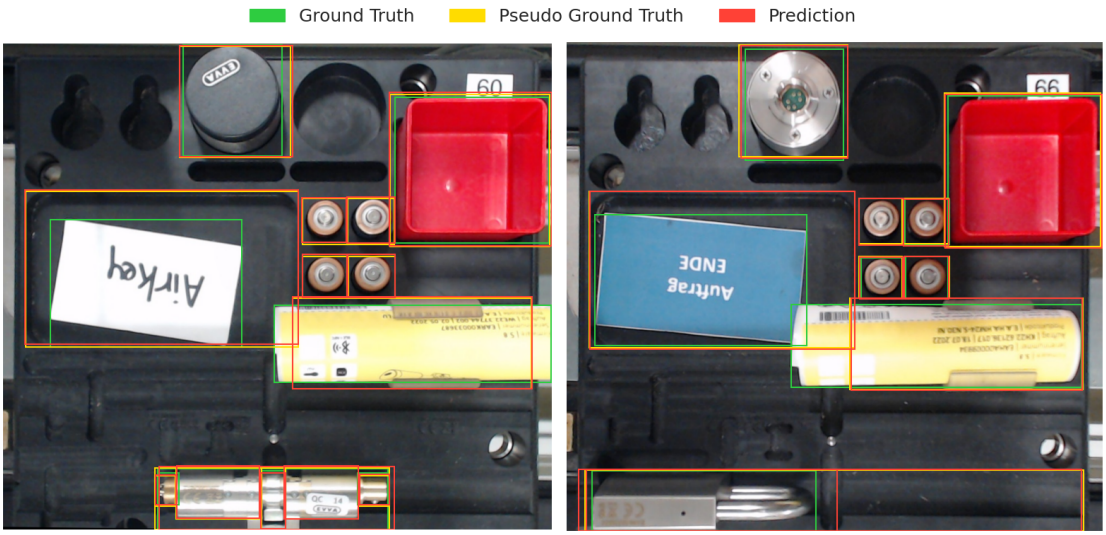


Figure 7.7: Examples of localization errors caused by inaccurate pseudo-annotations. The model replicates the pseudo ground truth for order notes but inherits their inaccuracy.

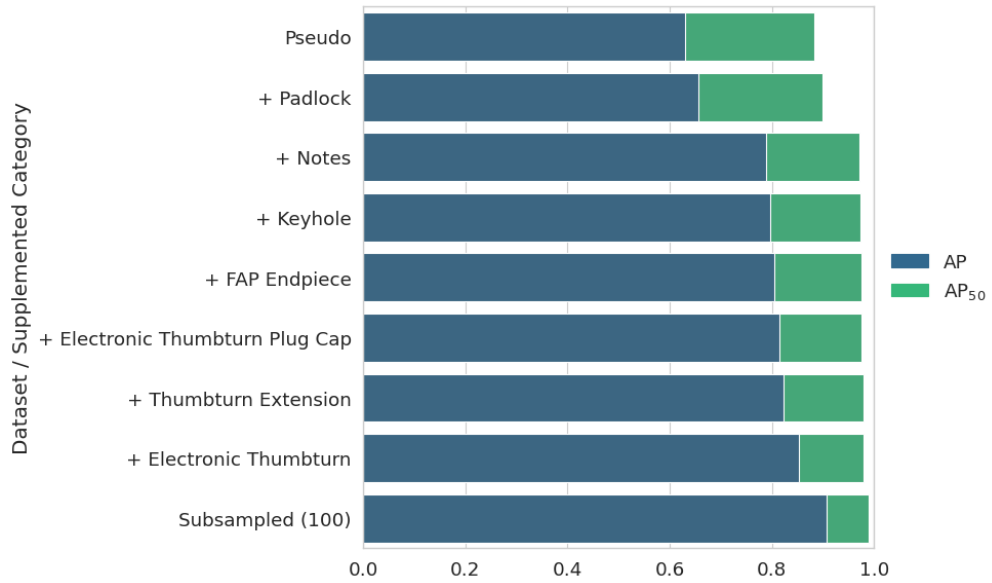


Figure 7.8: Visualization of the impact of supplemental annotations.

7.2.3 Synthetic Data Generation

This section will cover the performance of models trained using our synthetic data generation approach. We consider the same subsampled datasets as before and use them to generate a corresponding synthetic dataset of 1000 images each. Figure 7.9 shows the results in conjunction with those of the baseline datasets. It illustrates that models trained on synthetic data achieve superior AP and AP_{50} on smaller datasets, while matching the baseline on intermediate ones. Only on the dataset with at least 100 annotations for each class, the model trained with synthetic data falls behind very slightly. The performance differences are observed across all categories rather than being limited to a specific subset.

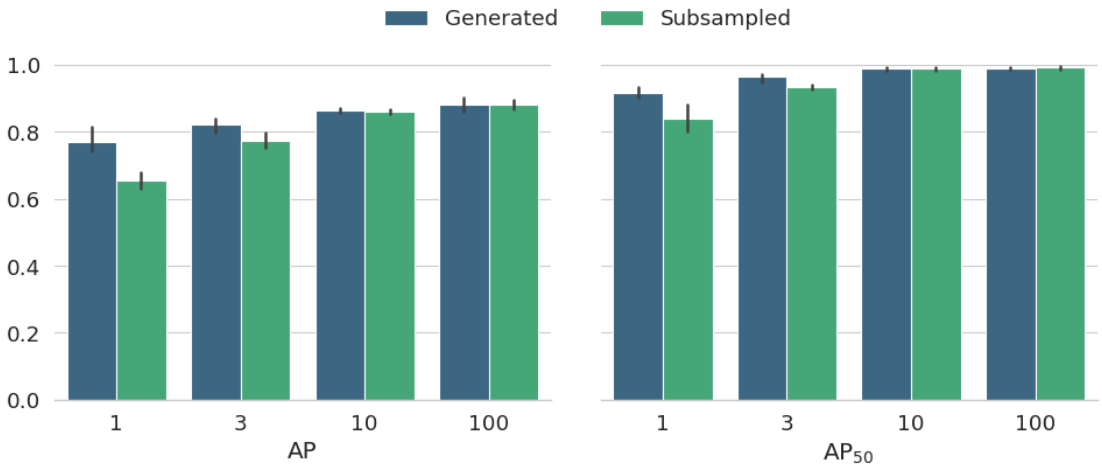
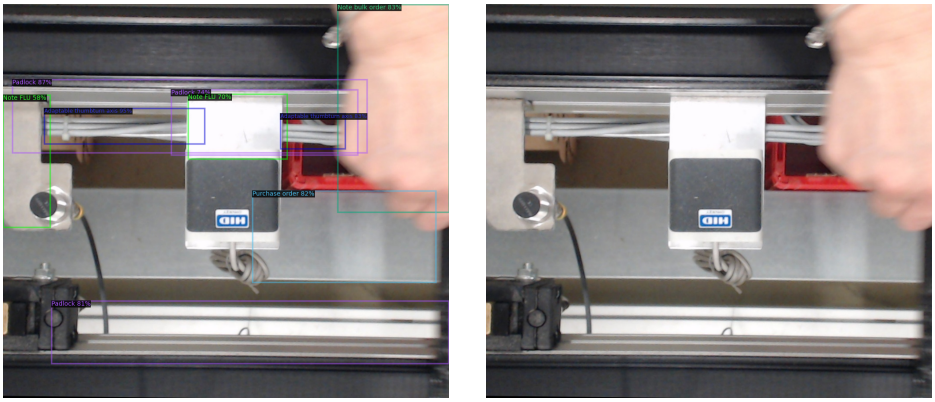


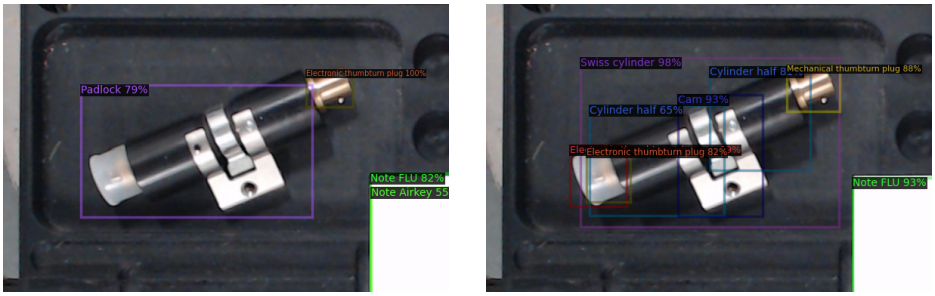
Figure 7.9: Performance comparison of models trained on synthetic and baseline datasets.

Evaluation on Misplaced Cylinders To demonstrate the benefits of synthetic training data in handling anomalous cases such as misplaced cylinders, we qualitatively compare the predictions of baseline models with those trained on synthetic data in Figure 7.10. All models were trained on the second smallest dataset, which contains at least three instances per class. The results shown generally coincide across different training folds. Note that these example images are not part of the training or validation datasets.

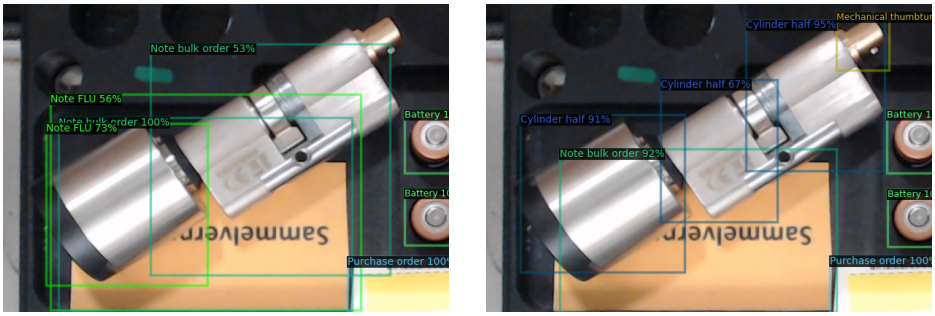
Figure 7.10a shows an example without a carrier. The standard model produces numerous false positives, particularly among rare classes, whereas the model trained with synthetic data does not exhibit these errors. In Figure 7.10b, a misplaced Swiss cylinder with a very rare black surface finish is shown. While the baseline configuration fails to detect the cylinder, the model trained with synthetic data succeeds, albeit the bounding boxes are somewhat imprecise. Figure 7.10c features a Euro profile cylinder already attached to its thumbturn, a configuration not seen in training. Both models incorrectly classify the thumbturn as a cylinder half. However, the version trained with synthetic examples is



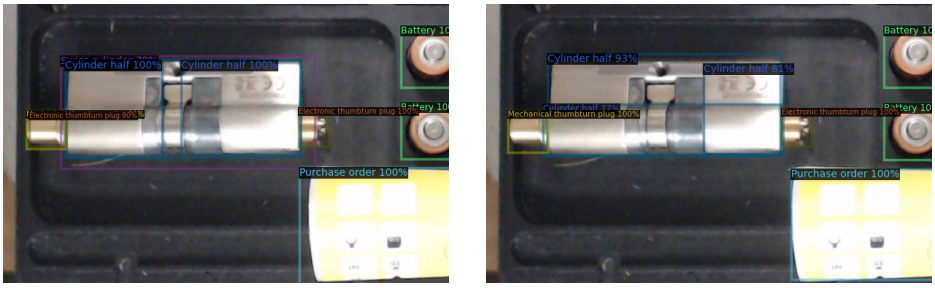
(a) Missing workpiece carrier



(b) Misplaced rotated Swiss round profile cylinder with rare surface finish



(c) Misplaced rotated Euro profile cylinder with attached thumbturn



(d) Upside-down cylinder

Figure 7.10: Comparison of predictions from models trained on a baseline dataset (left) and its synthetic counterpart (right).

able to detect most of the remaining components correctly. The cam remains undetected, although a model trained with increased rotational variation in the synthetic samples does capture it, at the cost of slightly reduced accuracy on more typical horizontal cylinders. Lastly, Figure 7.10d displays an upside-down cylinder, which neither the baseline nor the augmented model detects successfully. Addressing such cases may require additional augmentations or adapted data generation, but such changes come with trade-offs in performance on more common scenarios and have to be considered in relation to the frequency of such edge cases.

7.3 Comparison

Until now, ROI-based methods and object detection have been evaluated separately. This was due to several key differences between the two approaches. First, their annotation requirements differ: object detection requires fully annotated images with bounding boxes for all objects, while ROI-based methods can be trained on partially labelled images, since each region is considered independently. This allows training on datasets where only the rare classes or unique instances are annotated in an image, significantly reducing annotation effort. Accordingly, different subsampling strategies were used for low-data evaluation. Second, their outputs and most suitable evaluation metrics differ. Object detection produces a variable number of bounding boxes per image, while ROI-based models provide one prediction per predefined region. Finally, there are slight differences in the set of predicted components and properties. For instance, cylinder surface finish is only predicted in ROI-based classification, and certain components like Scandinavian and Rim cylinders are treated as single units in classification but are split into separate parts for detection.

To nonetheless enable a direct comparison, we retrain the classification and regression models using the exact same images as those used for training the object detection models. Performance is then evaluated solely based on region-based F1-scores, focusing on the subset of parts and properties that are comparable across both approaches. For object detection predictions, we apply class-specific confidence thresholds and non-maximum suppression (NMS) across classes assigned to the same region assuming that only one valid object can be present per region. Part length estimations are derived from the width of predicted bounding boxes and mapped to the corresponding discrete length categories.

The results, shown in Figure 7.11, demonstrate that all approaches generally perform well. ROI-based methods slightly outperform the baseline object detection model across nearly all regions, especially on smaller datasets. The only exception is a marginal advantage for object detection on the red tray region. Detection models trained with pseudo labels achieve results close to or on par with the baseline in most cases, and underperform only in the few problematic regions previously discussed. Results for models trained on synthetic data also align well with earlier findings: they surpass the baseline on smaller datasets, but tend to match or slightly trail the baseline on larger datasets.

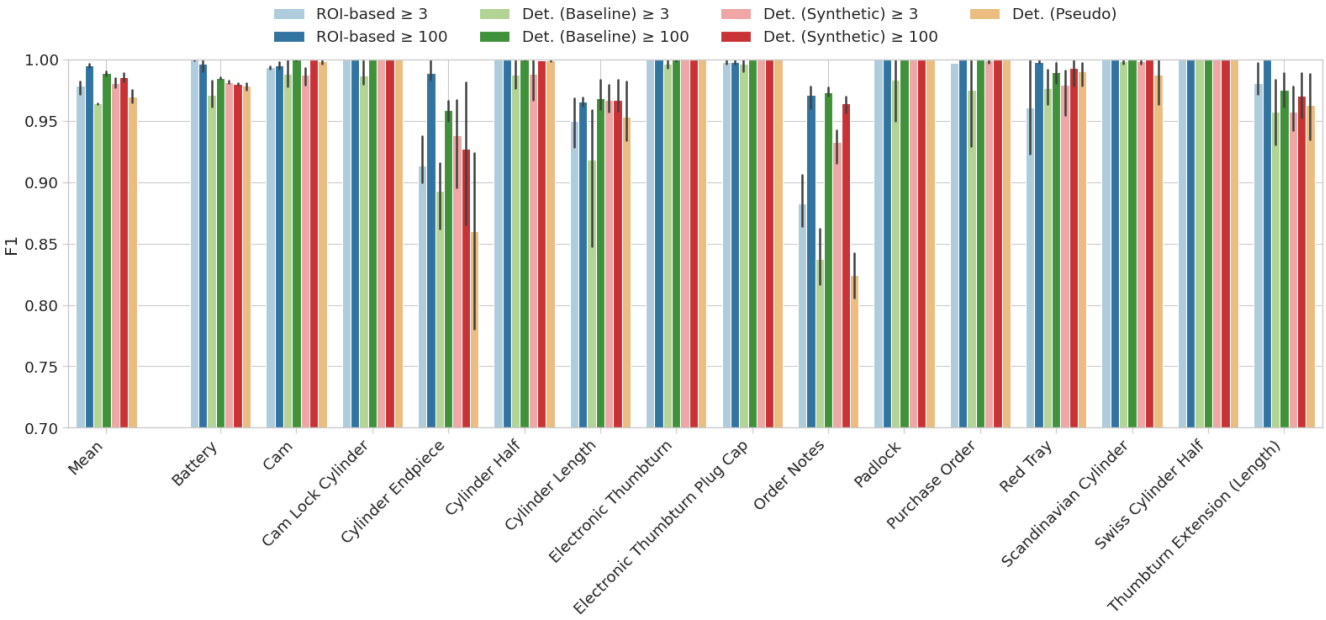


Figure 7.11: Comparison of ROI-based with object detection models. $\geq X$ signifies the dataset with a least X annotations per class.

The reduced annotation effort required for ROI-based methods makes their results even more favourable. However, their performance relies on the training data capturing all relevant visual variations. For instance, if a component from one region overlaps into another one and such cases are not represented in the training set, classification accuracy can deteriorate. Object detection does not share this limitation and is capable of identifying misplaced or rotated components. However, it still requires sufficient examples of such variations to detect them, particularly for rare or atypical appearances, like a battery lying on its side. Additionally, detection models not using synthetic data are more prone to false positives, especially when encountering components not seen during training as shown in Section 7.2.3.

7.4 Dataset Size and Class Balance

This section evaluates how model performance scales with dataset size and how increasing class imbalance affects ROI-based classification and object detection. For both methods, smaller datasets were created by subsampling the original, highly imbalanced training data. ROI-based classification datasets were constructed by selecting a fixed number of samples per class, whereas object detection datasets were built by ensuring a minimum number of annotations per class. Consequently, smaller datasets are more balanced, while larger ones increasingly reflect the original class distribution. This setup highlights both general data efficiency and the trade-offs between manual dataset curation, such

as targeted collection of under-represented classes, and passive collection during regular production.

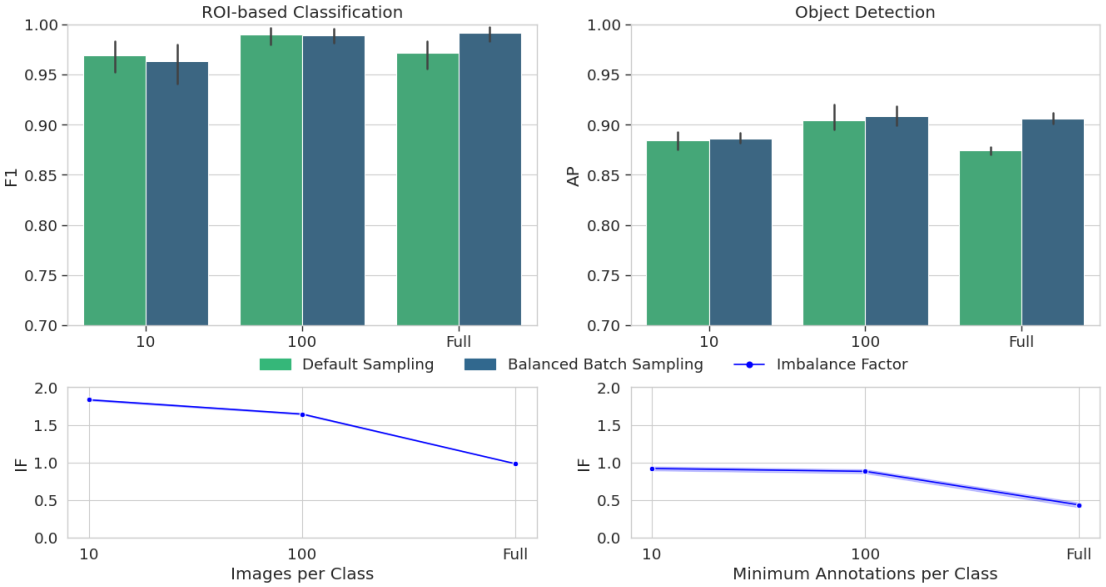


Figure 7.12: Average performance of models trained on datasets of varying size and class imbalance for ROI-based classification and object detection.

Previous results have shown that both ROI-based classification and object detection show large performance improvements up to the dataset with (at least) 10 samples per class. Afterwards, we can observe small improvements that are far less uniform than before since the source dataset only contains less than 100 instances for many classes. When using the full training set, the results diverge depending on the sampling strategy. As shown in Figure 7.12, with balanced batch sampling, performance generally stagnated, showing no substantial gains compared to the largest balanced subsampled dataset. Without class-aware sampling, a clear performance regression of approximately 4 percentage points was observed, caused by the dominance of frequent classes and configurations.

In conclusion, the number of instances per class is more important than the overall dataset size. A large number of samples from common classes provides little benefit in environments with low intra-class variance and can even lead to a regression in performance if class imbalance is not addressed. Passive data collection is a practical starting point, but it should be complemented by targeted sampling to capture the full range of operational conditions needed for reliable deployment.

CHAPTER 8

Discussion

This chapter summarizes the results of the previous chapter, compares the proposed methods, highlights limitations, and suggests directions for future work.

8.1 Comparison

This thesis explored three distinct strategies to enable data-efficient visual inspection through constrained component placement and low intra-class variance. Each approach presents a different trade-off between data requirements, annotation effort, model robustness, and applicability. This section provides a comparative analysis of these methods and outlines the scenarios in which each is most effective.

ROI-based methods proved to be the most accurate and data-efficient approach. The reformulation of detection into a set of localized classification tasks allows for training with only image-level labels. The method performed particularly well for tightly constrained regions, such as those for batteries and thumbturns, even with only a handful of images. However, a separate model must be trained for each region. While this complicates training, it also enables region-specific adaptations and fine-tuning. The main limitation lies in the assumption of correct component placement and consistent orientation, making it unsuitable for detecting misplaced or rotated parts. The severity of this drawback depends on the application context, as such anomalies typically result in false negatives, which are often more acceptable than incorrect classifications or false positives.

Object detection using pseudo annotations offers data efficiency on par with ROI-based methods, while providing greater flexibility. A single model is used to detect all components across the entire image, regardless of their position, thus making the approach better suited for identifying misaligned or misplaced parts. Pseudo bounding boxes derived from the constrained layout, reduce annotation effort while maintaining reasonable performance. The approach shares some of the same issues as ROI-based methods. While

it can detect misplaced components, the pseudo-annotations do not include such cases unless manually supplemented. Performance also tends to drop in loosely constrained regions, where pseudo bounding boxes are inaccurate. Additionally, object detection in general can struggle with visually similar classes, such as distinguishing between empty and full red trays. In these cases, dominant subcategories (empty tray) can overwhelm rarer ones (full tray) during training. As further measure such as two-stage or hybrid approaches as mentioned in Section 2.1.3 and Section 8.3 are needed to avoid such bias.

Unlike the previous two strategies, which focus primarily on reducing annotation effort, synthetic data generation aims to reduce the total number of real images required. A key advantage of this approach is its ability to intentionally simulate anomalies like rotated or misplaced components, making it highly effective for enhancing robustness to unpredictable real-world scenarios. However, this comes at the cost of realism: synthetic samples inherently risk domain mismatch compared to real-world images, a challenge amplified by our reliance on bounding box annotations instead of pixel masks. Lastly, designing a suitable generation pipeline that produces usable, realistic training images introduces additional complexity.

In summary, ROI-based methods offer the best overall results, making them the preferred choice in suitable applications. The strong data efficiency and ability to work with weak labels make them ideal for scenarios with constrained component placement and low intra-class variance. Pseudo bounding boxes provide similar efficiency, but are most effective if used to reduce annotation effort in object detection for a subset of classes. These can then be refined using manual ground truths or with other methods as needed. Synthetic data generation is most useful if collecting real samples is difficult, expensive, or if a large number of product variants or configurations exist. It is effective in handling anomalies and under-represented cases and could be combined with ROI-based methods to form a hybrid approach. Together, these methods support scalable, data-efficient inspection pipelines tailored to the demands of flexible and customized manufacturing.

8.2 Limitations

While the proposed methods demonstrate strong performance in the context of data-efficient visual assembly verification, several limitations must be acknowledged.

First, the scope of this evaluation is limited to a single use case, namely a specific electronic lock assembly line at EVVA. Nevertheless, we believe this environment to be representative of a relevant subclass of industrial inspection applications. This subclass includes scenarios such as PCB verification and aligns with numerous use cases discussed in Chapter 2. It remains unclear how well the findings generalize to other manufacturing domains, particularly those with less predictable layouts or broader visual variability. Applying the methods to other products or environments may uncover challenges not captured by this thesis.

Second, this thesis focuses on data-centric strategies. Although we optimize the model

and training, innovation is limited to dataset manipulation. Consequently, we did not investigate custom backbones, novel mechanisms, or task-specific modules. This decision was based on the unique opportunities identified in the data itself and the wide variability of architectures observed in related work. However, this choice inevitably limits the scope of insights, particularly with respect to object detection performance.

These limitations highlight the scope within which the proposed methods operate effectively and point toward necessary considerations for future research and broader applicability.

8.3 Future Work

This section outlines potential areas for future research.

Scalability and Data Deduplication While our approach has demonstrated strong performance on small datasets, its scalability to larger datasets is not yet optimal, particularly due to some classes showing performance regressions on larger datasets. To address this, both methods could benefit from data deduplication, by filtering out highly similar images. This should enhance training efficiency, especially for larger datasets that contain a large number of highly similar images. Implementing this in a self-supervised manner could also significantly reduce annotation effort. Additionally, the same similarity measure used for deduplication could be integrated into the data generation process to prioritize the most distinct and informative samples. As datasets grow larger, the techniques used to train models may also need to be adjusted.

High-Fidelity Synthetic Data Generation Regarding synthetic data generation, using a curated set of high-resolution images with pixel mask annotations could provide greater flexibility in recombination, enabling the creation of a large, high-quality dataset with minimal manual annotation effort. Ensuring that these images are free of occlusions would also increase the quality of the generated synthetic images. This approach would also make it feasible to use image segmentation methods without the need for manual time-intensive pixel mask annotations.

Pseudo-Annotation Refinement Next, leveraging class activation maps (CAMs) from ROI-based models or integrating weakly-supervised object localization (WSOL) techniques could help mitigate challenges associated with less constrained components while using pseudo bounding boxes. This approach would offer an alternative means of enhancing localization accuracy, without supplemental manual annotations.

Hybrid Approaches Another direction for future research is the development of hybrid approaches that combine object detection and ROI-based classification. Specifically, in cases where a compound part has an unconstrained position, such as in the one examined by Lin et al. [Lin+24], rotated object detection can first be used to localize the part.

Then, ROI-based methods can be used to perform fine-grained classification or property verification of the constituent parts within the detected region. This two-stage approach would broaden the applicability of ROI-based models to cases with variable component positioning, offering greater flexibility without solely relying on object detection. It is conceptually similar to the approach by Zhang et al. [Zha+24b], who detect arbitrarily placed saw chains using their rivets as reference points, but allows for more adaptable downstream analysis.

Shared Backbone for ROI-Based Methods While distinct models were trained for each region in this study, another possibility could involve using a shared backbone with region-specific classification heads. This would allow the model to share knowledge across regions, particularly in identifying relevant features in the given application, which could help reduce false positives and improve overall robustness.

Enhancing Robustness with Out-of-Distribution Backgrounds Another avenue for future work is to generalize the use of unrelated background images applied for synthetic data generation. These diverse backgrounds from a dataset like COCO could also be integrated into the training of other object detection and ROI-based models. We anticipate similar improvements in the handling of unexpected situations and a reduction in false positives.

CHAPTER 9

Conclusion

This thesis explored three data-centric strategies for enabling data-efficient deep learning in assembly verification: ROI-based classification and regression, pseudo-annotation based on constrained positions, and synthetic data generation through image composition. These approaches were evaluated on a real-world use case in the manufacturing of electronic locks, where traditional machine learning methods had previously failed to meet requirements.

The first strategy, ROI-based classification and regression, leverages the constrained spatial layout of components on the workpiece carrier to reformulate the detection problem into localized classifications. This method delivers respectable performance with only a handful of training images and achieves near-perfect accuracy if more data is used. Additionally, it requires only image-level labels, which are more readily available than bounding boxes or pixel mask annotations.

The second strategy involved reducing annotation effort through pseudo-annotations. By again leveraging the known geometry of the carrier tray, approximate bounding boxes were derived from image-level labels. This enabled the training of object detection models without costly manual annotation. While pseudo-annotations proved accurate enough for many components, they proved less useful for components with high positional variance.

The third strategy employed synthetic data generation through compositing. By reassembling real cropped component images into new configurations and placing them on varied backgrounds, this approach increased dataset diversity and improved model robustness, particularly for rare classes.

While effective in this context, the generalizability of these methods to other manufacturing settings remains to be explored. Future research may address scalability to larger datasets, improve synthetic and pseudo-labelling methods, or explore hybrid approaches that combine object detection with ROI-based methods. Enhancing model robustness

through the use of shared backbones and the inclusion of varied backgrounds also present promising directions.

In summary, this thesis demonstrates that data-efficient deep learning can effectively reduce dataset requirements. This is achieved by leveraging domain-specific characteristics such as constrained object locations and low intra-class variance.

APPENDIX A

Additional Results and Materials

A.1 Comparison of Single and Multi-Label Models

As detailed in Section 4.2, we classified cylinder endpieces using four distinct single-label models instead of a single multi-label model. The singular reason it is a multi-label problem, is due to the extension and dust cap of electronic thumbturn cylinder endpieces. This approach represents a trade-off between the effort of training multiple models and the potential for improved accuracy through specialization. As shown in Figure A.1, the aggregated results from the single-label models outperform the multi-label model across all dataset sizes by a small margin.

A.2 Alternative Learning Paradigms

All of the methods proposed in this thesis rely on supervised learning. At first glance, this may seem surprising given the growing emphasis on self- and weakly-supervised learning to reduce annotation requirements. Also, few-shot learning targets a different, but seemingly more relevant scenario: cases with only a small number of labelled examples per class. This makes it appear particularly well-suited to our setting. From this perspective, it is essential to evaluate how well these paradigms perform compared to our proposed supervised models. As such, we evaluated a selection of state-of-the-art (SOTA) approaches across these paradigms in Section 2.2 on both the region and the object detection datasets.

While we aim to provide a fair comparison by adapting each method to our dataset based on the insights gained during development of the proposed methods, conducting extensive hyperparameter optimization for every method is infeasible. Therefore, the results should be viewed as an indication of the general suitability of the corresponding learning paradigm and concepts rather than the optimal performance achievable.

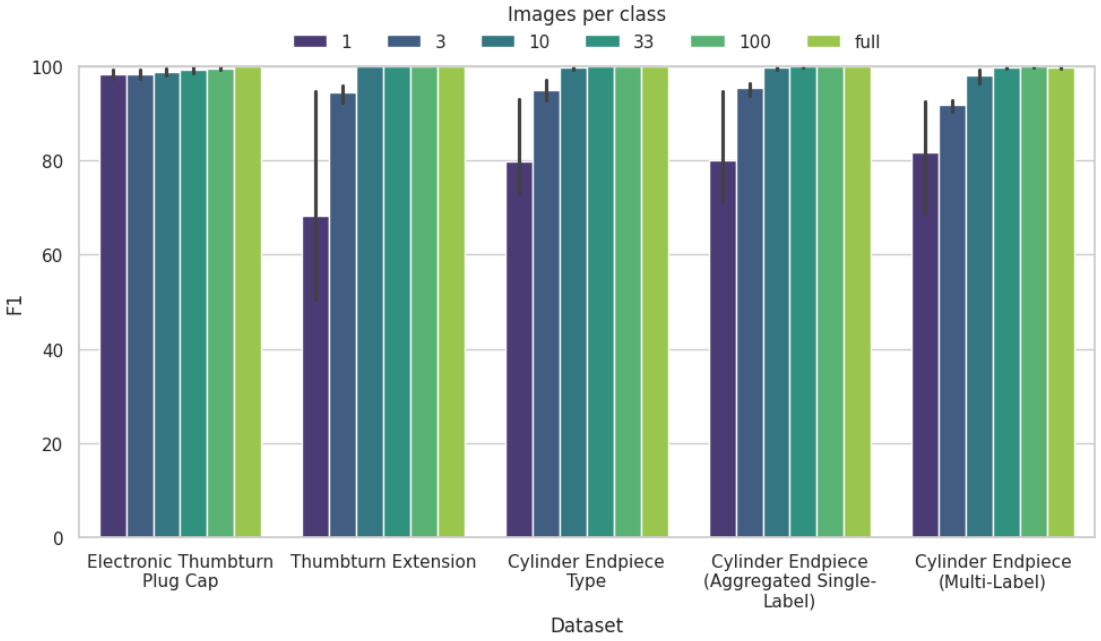


Figure A.1: Comparison of multiple single-label models to a single multi-label model for classifying cylinder endpieces across different dataset sizes. The aggregated single-label results reflect the combined performance of all single-label models on the multi-label dataset.

We provide an overview of selected state-of-the-art methods suitable for the region datasets in Table A.1, along with their corresponding results in Figure A.2. Among these, SimCLRv2, pretrained on the full dataset and fine-tuned with 10 images per class, yields the best performance after our approach. However, we evaluated this method on only a small subset of regions, as its self-supervised nature benefits from extended training. Next, few-shot classification methods perform well, with ProtoNet overall slightly outperforming CAML. Both were evaluated using 10 images per class. Despite this, their performance remains notably below the supervised and semi-supervised results. This aligns with findings from Nakamura and Harada [NH19] as well as those from Chen et al. [Che+19], which suggest that training a linear classifier or fine-tuning a pretrained model can match or surpass many meta-learning or few-shot approaches. TURTLE, which requires no training, was tested on the full dataset. It achieved the worst results, though it still demonstrates promising results in certain regions.

Figure A.3 and Table A.2, similarly give an overview of selected object detection methods applied to our dataset. Our proposed methods perform the best overall, showing a strong suitability for this dataset. The semi-supervised method, MixPL, follows closely behind, showing competitive results while using additional unlabelled data. Among the few-shot detection approaches, TFA achieves reasonable performance, but FSCE struggles, highlighting the challenges of few-shot object detection techniques.

Learning Paradigm	Method	Model	F1
Supervised	Ours	ResNeXt-50	0.9846
Self-Supervised Pretraining	SimCLRv2 [Che+20]	ResNeXt-50	0.7727[1]
		ResNet-18	0.7721[1]
Few-shot classification	ProtoNet [SSZ17]	CLIP (ViT-B/16)	0.7940[2]
		ResNet-34	0.8155[2]
	CAML [Fif+24]	CLIP (ViT-B/16)	0.7832[2]
		ResNet-34	0.7645[2]
Unsupervised clustering	TURTLE [GJB24]	CLIP (ViT-L/14) & DINOv2	0.4801[2]

Table A.1: Overview of selected state-of-the-art methods evaluated on the region datasets.
[1] Includes only results for order notes, cylinder endpieces and electronic thumbturns.
[2] Includes only results for single-label regions.

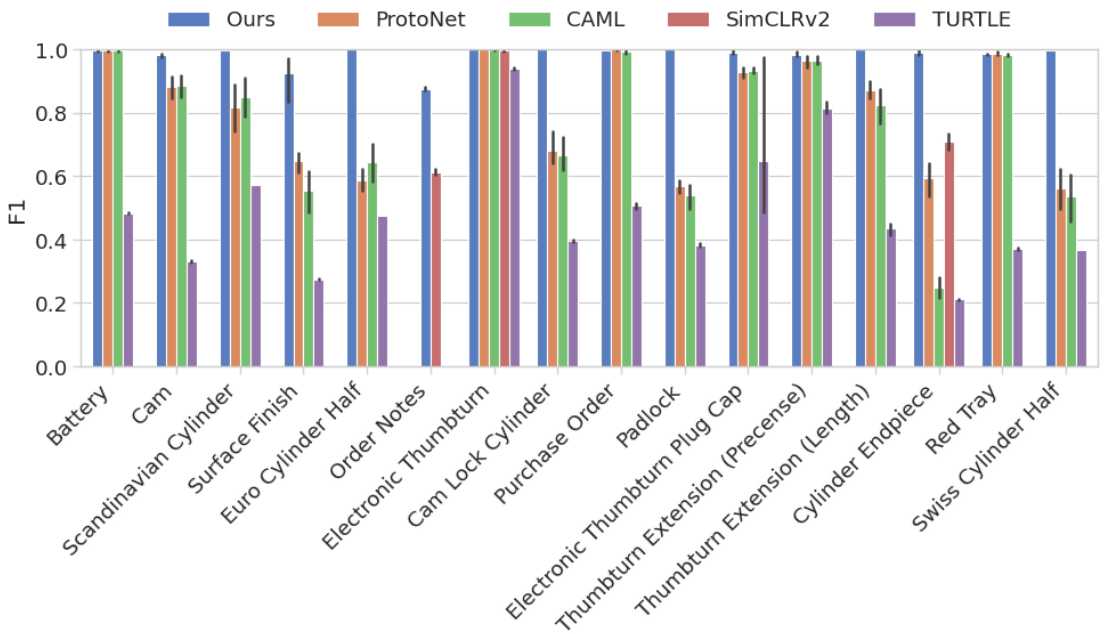


Figure A.2: Comparison of selected state-of-the-art approaches to ours on the regions datasets with 10 images per class.

Learning Paradigm	Method	Model	AP	AP_{50}
Supervised	Ours	Faster R-CNN (ResNet-50)	0.8860	0.9845
	Ours (Pseudo)	Faster R-CNN (ResNet-50)	0.8861	0.9845
	Ours (Synthetic)	Faster R-CNN (ResNet-50)	0.8620	0.9877
Semi-Supervised	MixPL [Che+23b]	Faster R-CNN (ResNet-50)	0.8636	0.9722
Few-shot detection	TFA [Wan+20]	Faster R-CNN (ResNet-50)	0.8335	0.9688
	FSCE [Sun+21]	Faster R-CNN (ResNet-101)	0.2879	0.4944

Table A.2: Comparison of selected SOTA approaches on the medium-sized dataset.

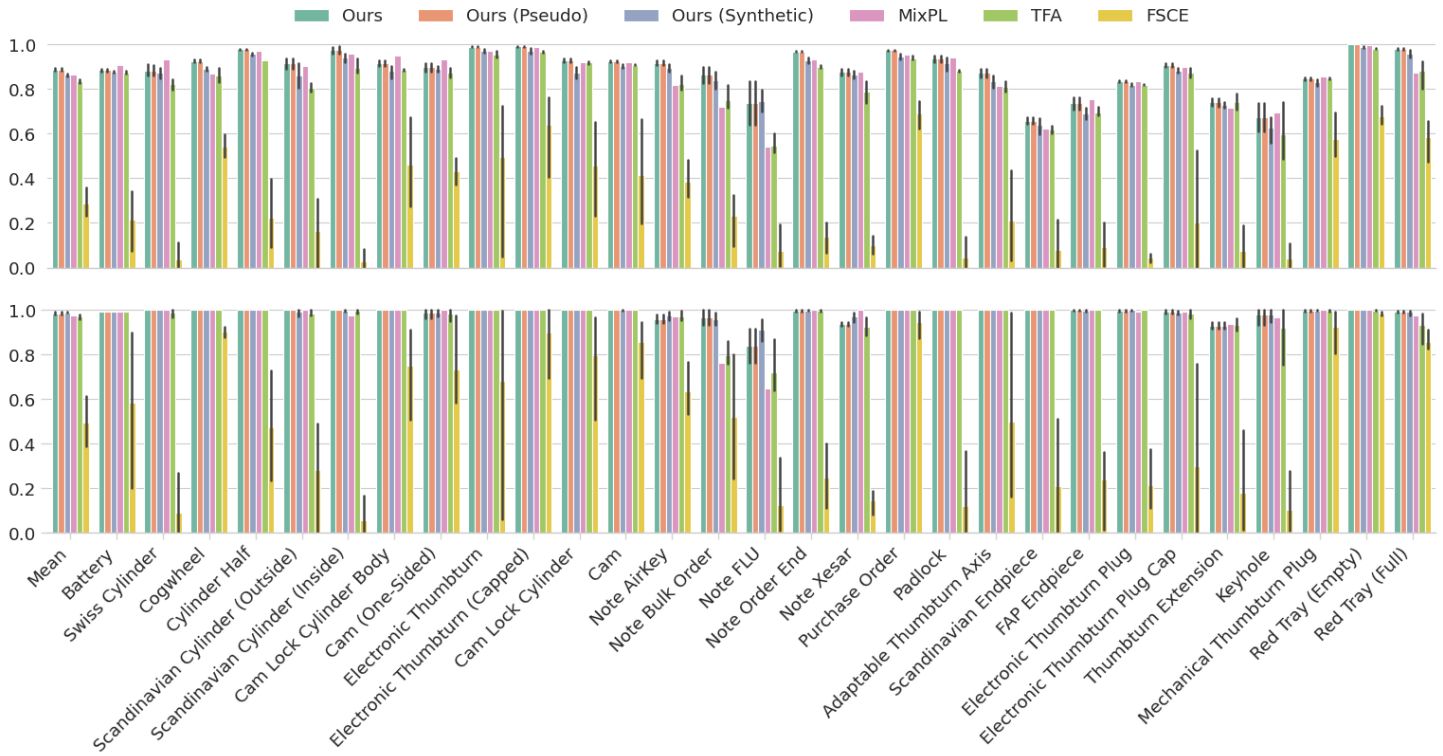


Figure A.3: Comparison of selected SOTA approaches on the medium-sized dataset.

A.3 Ablation Study

A.3.1 ROI-Based Methods

To support our choice of hyperparameters, augmentation and backbone, we also consider alternatives and variations in an ablation study. However, this study will be limited to the cylinder endpiece region with 10 images per class, as the number of training runs would otherwise become intractable. The results can be seen in Figure A.4.

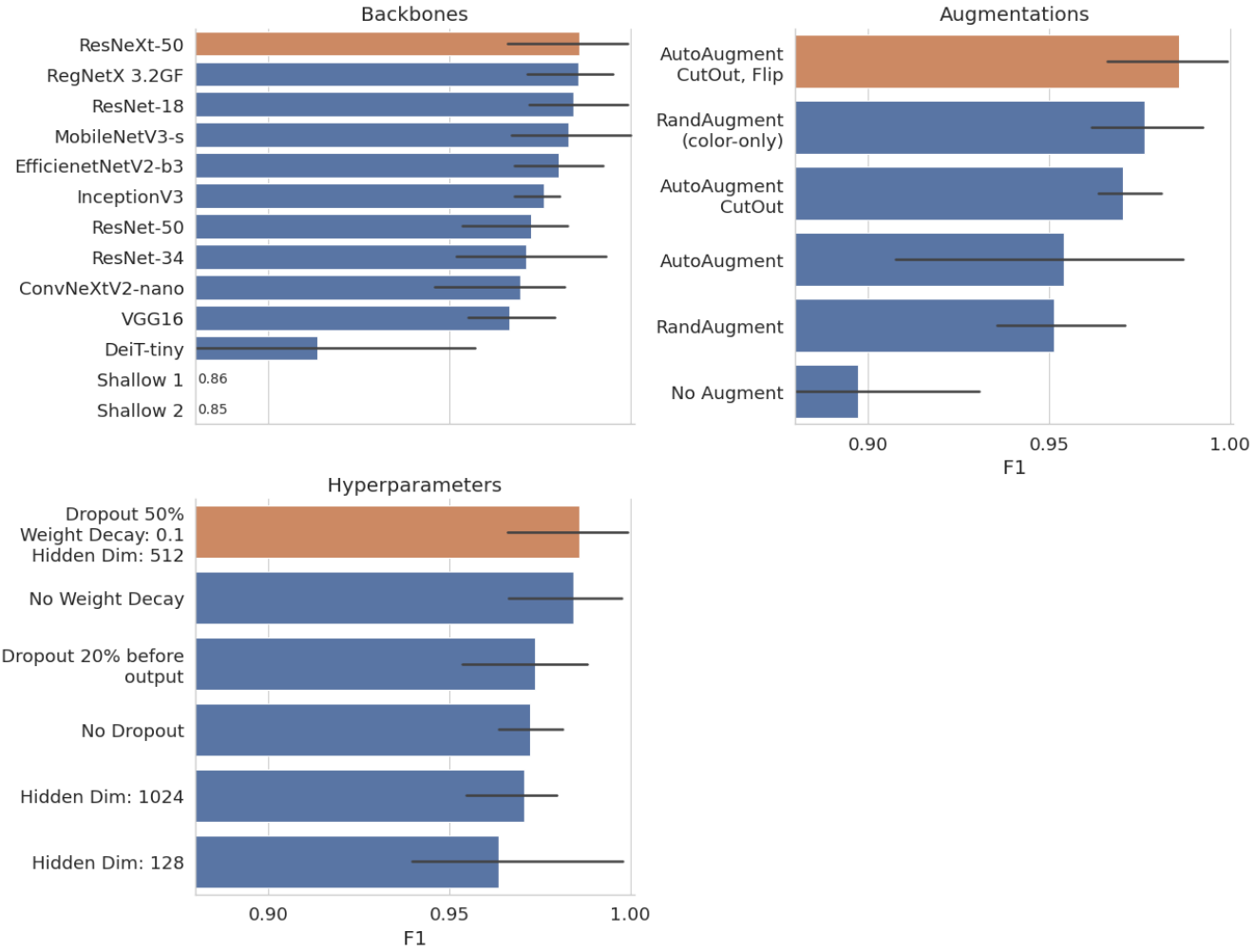


Figure A.4: Performance comparison of alternative backbones, augmentations and hyperparameters for cylinder endpiece classification with 10 images per class based on the F1-score.

In the backbone comparison (top left panel), most architectures performed comparably to the selected ResNeXt-50. Lightweight models such as MobileNetV3, ResNet-18, and EfficientNetV2 also yielded strong results, indicating their suitability for deployment in

resource-constrained environments. In contrast, shallow networks and the data-efficient transformer (DeiT) demonstrated significantly lower performance. The "Shallow 1" model consists of three stages, each comprising a convolutional layer, ReLU activation, and dropout, interleaved with max pooling. "Shallow 2" is constructed similarly but employs two convolutional layers per stage.

The augmentation analysis (top right panel) underscores the effectiveness of our augmentation strategy. Removing individual components, such as flipping or CutOut, leads to a decline in performance, while the absence of augmentations altogether results in the lowest F1-scores.

In the hyperparameter ablation study (bottom panel), performance variations are more subtle. The most impactful factor is the size of the hidden dimension in the classification head. Removing dropout causes a slight drop in performance, more pronounced on smaller datasets.

A.3.2 Object Detection

As with ROI-based classification we will now give a brief comparison of object detection with a selection of alternative models using different hyperparameters, feature extractors or augmentation pipelines in Figure A.5.

Overall, performance varied less than with ROI-based classification. The backbone selection was the most influential factor, although no distinct trend emerged for R-CNNs. Notably, DETR performed considerably worse, likely due to its requirement for larger datasets and extended training periods. Among augmentation strategies, incorporating Mosaic had the most substantial effect for this dataset, perhaps because it enhances the detection of features located at image borders. Hyperparameter choices had only a minor effect on overall performance. The small impact of OHEM is probably attributable to the balanced nature of this smaller dataset.

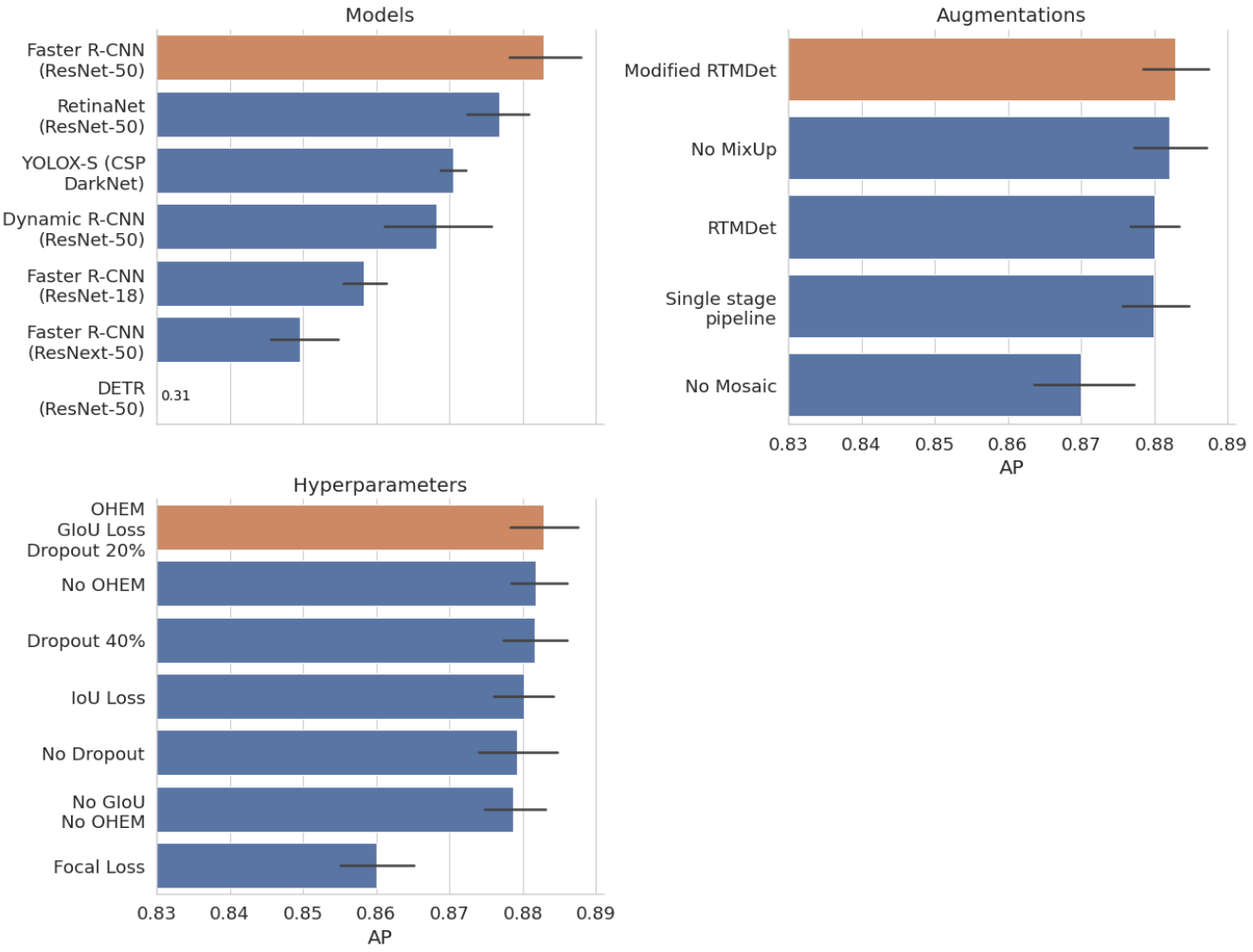


Figure A.5: Performance comparison of alternative backbones, augmentation pipelines, and hyperparameters based on AP for baseline object detection using the dataset with at least 10 annotations of each class.

A.4 Full Results Tables

Table A.3 presents the results of our proposed ROI-based method trained on datasets of varying sizes and evaluated on the test set. The corresponding object detection results are shown in Table A.4 (AP) and Table A.5 (AP_{50}). Both sets of results closely align with the previously reported validation performance, indicating minimal sample bias.

Region	**1**	**3**	**10**	**100**	**full**
Battery	99.4	99.4	99.7	99.9	99.9
Cam	76.9	95.2	98.1	98.7	99.9
Scandinavian Cylinder	90.2	99.3	99.7	100.0	100.0
Surface Finish	60.3	74.6	92.1	95.9	94.3
Euro Cylinder Half	50.4	99.1	100.0	100.0	100.0
Order Notes	22.6	45.5	28.7	30.7	19.6
Electronic Thumbturn	99.9	100.0	100.0	100.0	100.0
Cam Lock Cylinder	92.6	100.0	100.0	100.0	100.0
Purchase Order	82.7	83.0	74.6	74.3	82.8
Padlock	50.5	45.6	47.9	53.3	65.0
Electronic Thumbturn Plug Cap	53.5	95.2	98.9	99.5	100.0
Thumbturn Extension (Length)	45.3	73.0	98.7	99.0	99.7
Thumbturn Extension (Presence)	81.1	83.2	98.0	99.9	93.2
Cylinder Endpiece	74.9	90.3	95.4	98.8	90.0
Red Tray	92.6	95.6	99.3	100.0	100.0
Swiss Cylinder Half	77.4	98.2	100.0	99.8	100.0

Table A.3: Performance comparison of differently sized datasets based on F1-scores.

Minimum annotations per category	**1**		**3**		**10**		**100**		**full**	
	Base	**Syn.**	**Base**	**Syn.**	**Base**	**Syn.**	**Base**	**Syn.**	**Base**	**Pseudo**
Mean	65.3	77.7	79.2	84.0	87.9	86.7	90.0	88.2	89.9	61.8
Battery	85.1	88.2	87.3	87.5	88.3	87.8	89.5	88.3	89.4	87.3
Swiss Cylinder	82.8	86.5	84.1	90.7	89.9	91.2	93.9	91.6	93.1	85.3
Cogwheel	71.4	86.9	82.9	90.3	90.2	92.1	93.3	93.4	92.3	87.9
Cylinder Half	92.3	95.1	95.2	96.1	98.0	97.6	97.4	97.3	98.5	87.9
Scandinavian Cylinder (Outside)	76.9	87.0	82.6	90.9	91.9	91.5	95.7	96.2	97.2	87.6
Scandinavian Cylinder (Inside)	68.5	69.5	84.2	97.3	96.3	98.2	99.6	95.1	98.3	91.5
Cam Lock Cylinder Body	80.0	83.2	83.8	84.0	87.5	85.2	88.2	84.7	88.0	85.3
Cam (One-Sided)	77.3	86.8	87.6	87.8	90.0	90.6	91.7	91.2	90.9	85.5
Electronic Thumbturn	91.7	97.8	97.0	97.8	98.7	98.3	99.2	98.0	98.9	81.4
Electronic Thumbturn (Capped)	89.5	96.7	97.7	97.2	99.0	97.7	99.1	98.1	99.0	84.0
Cam Lock Cylinder	80.9	89.3	86.5	88.1	90.2	88.7	90.9	90.8	91.3	87.9
Cam	88.3	90.1	91.2	90.2	92.7	91.0	92.4	91.4	92.8	87.3
Note AirKey	73.1	86.5	81.0	89.2	90.6	91.7	93.3	92.4	93.4	4.7
Note Bulk Order	22.4	74.9	75.5	77.0	83.2	83.7	86.1	86.9	87.1	18.3
Note FLU	4.0	47.9	27.0	65.8	86.5	79.1	87.6	81.1	88.9	0.4
Note Order-End	88.1	97.0	95.3	96.9	97.4	97.4	98.4	97.4	98.1	22.5
Note Xesar	12.4	78.3	59.9	79.8	83.8	83.8	87.2	84.9	85.4	2.6
Purchase Order	91.1	94.1	95.4	94.5	97.6	94.8	97.4	94.8	97.0	44.2
Padlock	58.4	97.5	84.4	98.2	95.6	99.0	97.2	99.0	97.0	24.2
Adaptable Thumbturn Axis	73.3	83.3	82.5	83.0	88.1	81.1	89.9	90.6	89.3	70.5
Scandinavian Endpiece	35.6	54.0	51.5	60.3	63.9	61.6	80.7	79.7	80.1	62.5
FAP Endpiece	29.0	22.3	32.0	36.9	51.5	50.9	56.9	53.1	56.7	35.5
Electronic Thumbturn Plug	77.6	80.0	81.1	81.7	83.1	82.3	84.7	83.2	84.2	68.1
Electronic Thumbturn Plug Cap	71.8	57.3	86.3	86.3	90.2	89.6	91.7	91.8	91.8	65.4
Thumbturn Extension	34.1	41.2	68.6	69.1	75.5	74.0	75.9	73.9	83.2	56.3
Keyway	20.4	20.6	41.2	44.8	66.2	54.2	66.7	50.1	58.9	27.2
Mechanical Thumbturn Plug	77.0	81.7	82.1	82.9	84.4	84.6	86.3	84.4	87.1	76.5
Red Tray (Empty)	92.7	99.7	99.3	99.4	99.8	99.8	99.9	99.9	99.9	86.3
Red Tray (Full)	48.1	79.6	94.3	93.0	98.2	98.3	99.7	99.4	99.4	89.0

Table A.4: Test split performance of the proposed object detection methods trained on differently sized datasets (AP).

Minimum annotations per category	**1**		**3**		**10**		**100**		**full**	
	Base	**Syn.**	**Base**	**Syn.**	**Base**	**Syn.**	**Base**	**Syn.**	**Base**	**Pseudo**
Mean	83.2	91.4	93.9	97.1	98.9	98.4	98.9	98.6	99.0	87.2
Battery	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.3	99.0
Swiss Cylinder	99.8	99.5	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Cogwheel	94.5	98.4	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Cylinder Half	99.6	99.7	99.7	99.7	100.0	100.0	100.0	99.7	100.0	99.7
Scandinavian Cylinder (Outside)	98.9	99.2	96.7	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Scandinavian Cylinder (Inside)	83.9	74.2	89.1	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Cam Lock Cylinder Body	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Cam (One-Sided)	89.8	97.5	98.6	97.7	99.3	99.2	100.0	100.0	100.0	100.0
Electronic Thumbturn	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Electronic Thumbturn (Capped)	99.9	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Cam Lock Cylinder	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Cam	100.0	99.6	100.0	99.7	100.0	99.3	100.0	100.0	100.0	100.0
Note AirKey	86.3	93.5	91.6	97.6	94.8	97.1	95.0	99.8	96.8	24.8
Note Bulk Order	29.3	98.0	97.1	97.9	97.9	98.4	100.0	99.7	100.0	94.0
Note FLU	6.1	62.1	34.4	77.3	98.0	91.4	97.3	90.9	99.4	4.1
Note Order-End	96.1	99.9	99.5	100.0	100.0	100.0	100.0	100.0	100.0	99.7
Note Xesar	18.2	85.6	65.8	87.7	90.7	93.4	92.6	93.2	91.0	12.8
Purchase Order	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	98.7
Padlock	92.6	100.0	98.4	100.0	100.0	100.0	100.0	100.0	100.0	55.6
Adaptable Thumbturn Axis	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Scandinavian Endpiece	97.9	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
FAP Endpiece	72.7	59.6	88.4	96.6	99.4	100.0	97.8	99.2	96.7	81.0
Electronic Thumbturn Plug	98.2	99.2	99.6	100.0	99.9	100.0	100.0	100.0	100.0	100.0
Electronic Thumbturn Plug Cap	96.2	98.4	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Thumbturn Extension	58.6	70.8	89.6	89.5	92.8	92.2	92.6	92.6	99.9	87.4
Keyway	37.7	35.4	79.6	78.1	97.0	84.9	95.3	84.3	87.9	73.0
Mechanical Thumbturn Plug	99.9	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Red Tray (Empty)	99.6	99.9	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Red Tray (Full)	56.8	81.2	96.7	94.4	100.0	99.3	99.9	100.0	100.0	100.0

Table A.5: Test split performance of the proposed object detection methods trained on differently sized datasets (AP_{50}).

List of Figures

List of Tables

Bibliography

[AAJ24] Milad Ashourpour, Ghazaleh Azizpour, and Kerstin Johansen. "Real-Time Defect and Object Detection in Assembly Line: A Case for In-Line Quality Inspection". In: *Flexible Automation and Intelligent Manufacturing: Establishing Bridges for More Sustainable Manufacturing Systems*. FAIM. Ed. by Francisco J. G. Silva, António B. Pereira, and Raul D. S. G. Campilho. Cham: Springer Nature Switzerland, 2024, pp. 99–106. ISBN: 978-3-031-38241-3. DOI: 10.1007/978-3-031-38241-3_12.

[Al+22] Alaa S. AlWaisy et al. "Identifying Defective Solar Cells in Electroluminescence Images Using Deep Feature Representations". In: *PeerJ Computer Science* 8 (May 19, 2022), e992. ISSN: 2376-5992. DOI: 10.7717/peerj-cs.992.

[Ara+24] Kerem Aras et al. "Automated Optical Inspection for Quality Control in PCBA Assembly Lines: A Case Study for Point of Sale Devices Production Lines". In: *International Congress on Human-Computer Interaction, Optimization and Robotic Applications*. HORA. Istanbul, Turkey: IEEE, May 23, 2024, pp. 1–8. ISBN: 979-8-3503-9463-4. DOI: 10.1109/HORA61326.2024.10550768.

[Arn97] Francis E. Arnstein. "Catalogue of Human Error". In: *British Journal of Anaesthesia* 79.5 (Nov. 1997), pp. 645–656. DOI: 10.1093/bja/79.5.645.

[BCV13] Yoshua Bengio, Aaron C. Courville, and Pascal Vincent. "Representation Learning: A Review and New Perspectives". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8 (2013), pp. 1798–1828. DOI: 10.1109/TPAMI.2013.50.

[BEM21] Julen Balzategui, Luka Eciolaza, and Daniel Maestro-Watson. "Anomaly Detection and Automatic Labeling for Solar Cell Quality Inspection Based on Generative Adversarial Network". In: *Sensors* 21.13 (June 25, 2021), p. 4361. ISSN: 1424-8220. DOI: 10.3390/s21134361.

[Ben+21] Tajeddine Benbarrad et al. "Intelligent Machine Vision Model for Defective Product Inspection Based on Machine Learning". In: *Journal of Sensor and Actuator Networks* 10.1 (Jan. 28, 2021), p. 7. ISSN: 2224-2708. DOI: 10.3390/jsan10010007.

[Car+18] Marc-André Carbonneau et al. “Multiple Instance Learning: A Survey of Problem Characteristics and Applications”. In: *Pattern Recognition* 77 (May 2018), pp. 329–353. ISSN: 0031-3203. DOI: 10.1016/j.patcog.2017.10.009.

[Che+19] Wei-Yu Chen et al. “A Closer Look at Few-shot Classification”. 2019. arXiv: 1904.04232 [cs]. Pre-published.

[Che+20] Ting Chen et al. “A Simple Framework for Contrastive Learning of Visual Representations”. In: *Proceedings of the 37th International Conference on Machine Learning (ICML)*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 1597–1607. DOI: 10.5555/3524938.3525087.

[Che+23a] Xing Chen et al. “A Comprehensive Review of Deep Learning-Based PCB Defect Detection”. In: *IEEE Access* 11 (2023), pp. 139017–139038. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2023.3339561.

[Che+23b] Zeming Chen et al. “Mixed Pseudo Labels for Semi-Supervised Object Detection”. Dec. 12, 2023. arXiv: 2312.07006. (Visited on 09/30/2024). Pre-published.

[Coc77] William Gemmell Cochran. *Sampling Techniques*. 3rd ed. Wiley Series in Probability and Statistics. Nashville, TN, USA: John Wiley & Sons, July 1977. 428 pp. ISBN: 978-0-471-16240-7.

[Cub+19] Ekin D. Cubuk et al. “AutoAugment: Learning Augmentation Strategies From Data”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA: IEEE, June 2019, pp. 113–123. ISBN: 978-1-7281-3293-8. DOI: 10.1109/CVPR.2019.00020.

[Cub+20] Ekin D. Cubuk et al. “RandAugment: Practical Automated Data Augmentation with a Reduced Search Space”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. Seattle, WA, USA: IEEE, June 2020, pp. 3008–3017. ISBN: 978-1-7281-9360-1. DOI: 10.1109/CVPRW50498.2020.00359.

[Czi+20] Tamás Czimmermann et al. “Visual-Based Defect Detection and Classification Approaches for Industrial Applications - A Survey”. In: *Sensors* 20.5 (Mar. 6, 2020), p. 1459. ISSN: 1424-8220. DOI: 10.3390/s20051459.

[Den+09] Jia Deng et al. “ImageNet: A Large-Scale Hierarchical Image Database”. In: *Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[DLZ24] Li Deng, Gang Liu, and Yilian Zhang. “A Review of Machine Vision Applications in Aerospace Manufacturing Quality Inspection”. In: *4th International Conference on Computer, Control and Robotics (ICCCR)*. Shanghai, China: IEEE, Apr. 19, 2024, pp. 31–39. ISBN: 979-8-3503-7314-1. DOI: 10.1109/ICCCR61138.2024.10585378.

[DS83] Colin G. Drury and Murray A. Sinclair. “Human and Machine Performance in an Inspection Task”. In: *Human Factors: The Journal of the Human Factors and Ergonomics Society* 25.4 (Aug. 1983), pp. 391–399. ISSN: 0018-7208. DOI: 10.1177/001872088302500404.

[DT17] Terrance DeVries and Graham W. Taylor. “Improved Regularization of Convolutional Neural Networks with Cutout”. Nov. 29, 2017. arXiv: 1708.04552 [cs]. Pre-published.

[Fif+24] Christopher Fifty et al. “Context-Aware Meta-Learning”. In: *International Conference on Learning Representations (ICLR)*. Jan. 16, 2024.

[FS10] George Forman and Martin Scholz. “Apples-to-Apples in Cross-Validation Studies: Pitfalls in Classifier Performance Measurement”. In: *ACM SIGKDD Explorations Newsletter* 12.1 (Nov. 9, 2010), pp. 49–57. ISSN: 1931-0145. DOI: 10.1145/1882471.1882479.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. Adaptive Computation and Machine Learning. Cambridge, Mass: The MIT press, 2016. ISBN: 978-0-262-03561-3.

[GDG19] Agrim Gupta, Piotr Dollár, and Ross B. Girshick. “LVIS: A Dataset for Large Vocabulary Instance Segmentation”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA: IEEE, 2019, pp. 5356–5364. DOI: 10.1109/CVPR.2019.00550.

[Ge+21] Zheng Ge et al. “YOLOX: Exceeding YOLO Series in 2021”. Aug. 6, 2021. arXiv: 2107.08430 [cs]. Pre-published.

[Geh+17] Jonas Gehring et al. “A Convolutional Encoder Model for Neural Machine Translation”. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*. Ed. by Regina Barzilay and Min-Yen Kan. Vancouver, Canada: Association for Computational Linguistics, 2017, pp. 123–135. DOI: 10.18653/V1/P17-1012.

[Gha+24] Hassan Gharoun et al. “Meta-Learning Approaches for Few-Shot Learning: A Survey of Recent Advances”. In: *ACM Computing Surveys* 56.12 (Dec. 31, 2024), pp. 1–41. ISSN: 0360-0300. DOI: 10.1145/3659943.

[Gir15] Ross Girshick. “Fast R-CNN”. In: *IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 1440–1448. DOI: 10.1109/ICCV.2015.169.

[GJB24] Artyom Gadetsky, Yulun Jiang, and Maria Brbi. “Let Go of Your Labels with Unsupervised Transfer”. In: *Proceedings of the 41st International Conference on Machine Learning (ICML)*. Vienna, Austria, 2024. DOI: 10.5555/3692070.3692645.

[He+22] Kaiming He et al. “Masked Autoencoders Are Scalable Vision Learners”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. New Orleans, LA, USA: IEEE, 2022, pp. 15979–15988. DOI: 10.1109/CVPR52688.2022.01553.

[He+23] Xiangjie He et al. “DG-GAN: A High Quality Defect Image Generation Method for Defect Detection”. In: *Sensors* 23.13 (June 26, 2023), p. 5922. ISSN: 1424-8220. DOI: 10.3390/s23135922.

[How+17] Andrew G. Howard et al. “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications”. Apr. 17, 2017. arXiv: 1704.04861 [cs]. Pre-published.

[Hüt+24] Nils Hütten et al. “Deep Learning for Automated Visual Inspection in Manufacturing and Maintenance: A Survey of Open- Access Papers”. In: *Applied System Innovation* 7.1 (Jan. 22, 2024), p. 11. ISSN: 2571-5577. DOI: 10.3390/asi7010011.

[HWZ19] Ming Han, Qingxiang Wu, and Xiongjun Zeng. “Single-Scale Workpiece Defect Detection Based on Deep Learning”. In: *12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*. IEEE, Oct. 2019. DOI: 10.1109/cisp-bmei48845.2019.8965923.

[JB23] Shashi Bhushan Jha and Radu F. Babiceanu. “Deep CNN-based Visual Defect Detection: Survey of Current Literature”. In: *Computers in Industry* 148 (June 2023), p. 103911. ISSN: 0166-3615. DOI: 10.1016/j.compind.2023.103911.

[KB20] Nejc Kozamernik and Drago Braun. “Visual Inspection System for Anomaly Detection on KTL Coatings Using Variational Autoencoders”. In: *53rd CIRP Conference on Manufacturing Systems* 93 (2020), pp. 1558–1563. ISSN: 2212-8271. DOI: 10.1016/j.procir.2020.04.114.

[KCT20] Ziqiu Kang, Cagatay Catal, and Bedir Tekinerdogan. “Machine Learning Applications in Production Lines: A Systematic Literature Review”. In: *Computers & Industrial Engineering* 149 (Nov. 2020), p. 106773. ISSN: 0360-8352. DOI: 10.1016/j.cie.2020.106773.

[KD19] Vijay Kotu and Bala Deshpande. “Anomaly Detection”. In: *Data Science*. Ed. by Vijay Kotu and Bala Deshpande. Second Edition. Morgan Kaufmann, 2019, pp. 447–465. ISBN: 978-0-12-814761-0. DOI: 10.1016/B978-0-12-814761-0.00013-7.

[KH06] Andreas M. Kaplan and Michael Haenlein. “Toward a Parsimonious Definition of Traditional and Electronic Mass Customization”. In: *Journal of Product Innovation Management* 23.2 (Mar. 2006), pp. 168–182. ISSN: 0737-6782. DOI: 10.1111/j.1540-5885.2006.00190.x.

[Koh95] Ron Kohavi. “A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection”. In: *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*. IJCAI’95. Montreal, Quebec, Canada: Morgan Kaufmann, 1995, pp. 1137–1143. ISBN: 1-55860-363-8. DOI: 10.5555/1643031.1643047.

[Kon+20] Qiuqiang Kong et al. “PANNs: Large-scale Pretrained Audio Neural Networks for Audio Pattern Recognition”. In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 28 (2020), pp. 2880–2894. DOI: 10.1109/taslp.2020.3030497.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. Ed. by F. Pereira et al. Vol. 25. Curran Associates, Inc., 2012. DOI: 10.1145/3065386.

[Kur+20] Ivan Kuric et al. “Visual Product Inspection Based on Deep Learning Methods”. In: *Advanced Manufacturing Processes*. InterPartner 2019. Ed. by Volodymyr Tonkonogyi et al. Cham: Springer, 2020, pp. 148–156. ISBN: 978-3-030-40724-7. DOI: 10.1007/978-3-030-40724-7_15.

[KV15] Agnieszka Kujawiska and Katarzyna Vogt. “Human Factors in Visual Quality Control”. In: *Management and Production Engineering Review* 6.2 (June 1, 2015), pp. 25–31. ISSN: 2082-1344. DOI: 10.1515/mper-2015-0013.

[KV94] Anders Krogh and Jesper Vedelsby. “Neural Network Ensembles, Cross Validation, and Active Learning”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. Ed. by G. Tesauro, D. Touretzky, and T. Leen. Vol. 7. MIT Press, 1994. URL: https://proceedings.neurips.cc/paper_files/paper/1994/file/b8c37e33defde51cf91e1e03e51657da-Paper.pdf.

[KZ01] Gary King and Langche Zeng. “Logistic Regression in Rare Events Data”. In: *Political Analysis* 9 (2001), pp. 137–163. DOI: 10.1093/oxfordjournals.pan.a004868.

[Lee+23] Xian Yeow Lee et al. “XDNet: A Few-Shot Meta-Learning Approach for Cross-Domain Visual Inspection”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. Vancouver, BC, Canada: IEEE, June 2023, pp. 4375–4384. ISBN: 979-8-3503-0249-3. DOI: 10.1109/CVPRW59228.2023.00460.

[Lin+14] Tsung-Yi Lin et al. “Microsoft COCO: Common Objects in Context”. In: *Computer Vision - ECCV 2014*. Ed. by David Fleet et al. Cham: Springer, 2014, pp. 740–755. ISBN: 978-3-319-10602-1. DOI: 10.1007/978-3-319-10602-1_48.

[Lin+24] Hong-Dar Lin et al. “Utilizing Deep Learning for Defect Inspection in Hand Tool Assembly”. In: *Sensors* 24.11 (June 4, 2024), p. 3635. ISSN: 1424-8220. DOI: 10.3390/s24113635.

[Liu+16] Wei Liu et al. “SSD: Single Shot MultiBox Detector”. In: *Computer Vision - ECCV 2016*. Ed. by Bastian Leibe et al. Cham: Springer, 2016, pp. 21–37. ISBN: 978-3-319-46448-0. DOI: 10.1007/978-3-319-46448-0_2.

[Liu+19] Guixiong Liu et al. “Chassis Assembly Detection and Identification Based on Deep Learning Component Instance Segmentation”. In: *Symmetry* 11.8 (Aug. 3, 2019), p. 1001. ISSN: 2073-8994. DOI: 10.3390/sym11081001.

[LKP19] Dae-ui Lim, Young-Gyu Kim, and Tae-Hyoung Park. “SMD Classification for Automated Optical Inspection Machine Using Convolution Neural Network”. In: *Third IEEE International Conference on Robotic Computing (IRC)*. Naples, Italy: IEEE, Feb. 2019, pp. 395–398. ISBN: 978-1-5386-9245-5. DOI: 10.1109/IRC.2019.00072.

[LXJ23] Chao Li, Wuhao Xia, and ZhengLiang Jiang. “Weak Feature Defect Generation with GAN for Faster RCNN Based PCB Defect Detection”. In: *8th International Conference on Data Science in Cyberspace (DSC)*. Hefei, China: IEEE, Aug. 18, 2023, pp. 306–312. ISBN: 979-8-3503-3103-5. DOI: 10.1109/DSC59305.2023.00051.

[Lyu+22] Chengqi Lyu et al. “RTMDet: An Empirical Study of Designing Real-Time Object Detectors”. Dec. 16, 2022. arXiv: 2212.07784 [cs]. Pre-published.

[Maz+20] Muriel Mazzetto et al. “Deep Learning Models for Visual Inspection on Automotive Assembling Line”. In: *International Journal of Advanced Engineering Research and Science* 7.3 (2020), pp. 473–494. ISSN: 2349-6495. DOI: 10.22161/ijaers.74.56.

[Met+19] Maximilian Metzner et al. “Automated Optical Inspection of Soldering Connections in Power Electronics Production Using Convolutional Neural Networks”. In: *9th International Electric Drives Production Conference (EDPC)*. Esslingen, Germany: IEEE, Dec. 2019, pp. 1–6. ISBN: 978-1-7281-4319-4. DOI: 10.1109/EDPC48408.2019.9011820.

[Mig20] Pedro Miguel. “Detection of Production Defects Using Machine Learning Based Image Classification Algorithms”. MA thesis. Porto: University of Porto, Sept. 1, 2020. URL: https://repositorio-aberto.up.pt/bitstream/10216/129867/2/427508.pdf.

[MTM22] Robert F. Maack, Hasan Tercan, and Tobias Meisen. “Deep Learning Based Visual Quality Inspection for Industrial Assembly Line Production Using Normalizing Flows”. In: *IEEE 20th International Conference on Industrial Informatics (INDIN)*. Perth, Australia: IEEE, July 25, 2022, pp. 329–334. ISBN: 978-1-7281-7568-3. DOI: 10.1109/INDIN51773.2022.9976097.

[NH19] Akihiro Nakamura and Tatsuya Harada. “Revisiting Fine-tuning for Few-shot Learning”. Oct. 3, 2019. arXiv: 1910.00216 [cs]. Pre-published.

[Nik21] Sergey I. Nikolenko. *Synthetic Data for Deep Learning*. Vol. 174. Springer Optimization and Its Applications. Cham: Springer International Publishing, 2021. ISBN: 978-3-030-75177-7. DOI: 10.1007/978-3-030-75178-4.

[NS21] Vahid Nasir and Farrokh Sassani. “A Review on Deep Learning in Machining and Tool Monitoring: Methods, Opportunities, and Challenges”. In: *International Journal of Advanced Manufacturing Technology* 115.9–10 (Aug. 2021), pp. 2683–2709. ISSN: 0268-3768. DOI: 10.1007/s00170-021-07325-7.

[OK19] Ridvan Ozdemir and Mehmet Koc. “A Quality Control Application on a Smart Factory Prototype Using Deep Learning Methods”. In: *IEEE 14th International Conference on Computer Sciences and Information Technologies (CSIT)*. IEEE, Sept. 2019. DOI: 10.1109/stc-csit.2019.8929734.

[PG22] Jo Plested and Tom Gedeon. “Deep Transfer Learning for Image Classification: A Survey”. May 20, 2022. arXiv: 2205.09904 [cs]. Pre-published.

[Qui+22] Joaquin Quinonero-Candela et al. *Dataset Shift in Machine Learning*. Neural Information Processing. The MIT Press, June 22, 2022. ISBN: 978-0-262-54587-7.

[Ren+17] Shaoqing Ren et al. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.6 (June 1, 2017), pp. 1137–1149. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2016.2577031.

[Rez+19] Hamid Rezatofighi et al. “Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA: IEEE, June 2019, pp. 658–666. ISBN: 978-1-7281-3293-8. DOI: 10.1109/CVPR.2019.00075.

[RL18] Leonard Rusli and Anthony Luscher. “Fastener Identification and Assembly Verification via Machine Vision”. In: *Assembly Automation* 38.1 (Jan. 23, 2018), pp. 1–9. ISSN: 0144-5154. DOI: 10.1108/AA-08-2016-093.

[RMM20] Oliver Rippel, Maximilian Müller, and Dorit Merhof. “GAN-based Defect Synthesis for Anomaly Detection in Fabrics”. In: *25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. Vienna, Austria: IEEE, Sept. 2020, pp. 534–540. ISBN: 978-1-7281-8956-7. DOI: 10.1109/ETFA46521.2020.9212099.

[Rus+15] Olga Russakovsky et al. “ImageNet Large Scale Visual Recognition Challenge”. In: *International Journal of Computer Vision* 115.3 (Dec. 2015), pp. 211–252. ISSN: 0920-5691. DOI: 10.1007/s11263-015-0816-y.

[Sel+20] Ramprasaath R. Selvaraju et al. “Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization”. In: *International Journal of Computer Vision* 128.2 (Feb. 2020), pp. 336–359. ISSN: 0920-5691. DOI: 10.1007/s11263-019-01228-7.

[SGG16] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. “Training Region-Based Object Detectors with Online Hard Example Mining”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA: IEEE, June 2016, pp. 761–769. ISBN: 978-1-4673-8851-1. DOI: 10.1109/CVPR.2016.89.

[Sha+24] Masoud Shaloo et al. “Flexible Automation of Quality Inspection in Parts Assembly Using CNN-based Machine Learning”. In: *Procedia Computer Science* 232 (2024), pp. 2921–2932. ISSN: 1877-0509. DOI: 10.1016/j.procs.2024.02.108.

[SLH16] Li Shen, Zhouchen Lin, and Qingming Huang. “Relay Backpropagation for Effective Learning of Deep Convolutional Neural Networks”. In: *Computer Vision - ECCV 2016*. Ed. by Bastian Leibe et al. Vol. 9911. Lecture Notes in Computer Science. Springer, 2016, pp. 467–482. DOI: 10.1007/978-3-319-46478-7_29.

[Son+24a] Jie Song et al. “A Fault Detection Method for Transmission Line Components Based on Synthetic Dataset and Improved YOLOv5”. In: *International Journal of Electrical Power & Energy Systems* 157 (June 2024), p. 109852. ISSN: 0142-0615. DOI: 10.1016/j.ijepes.2024.109852.

[Son+24b] Zhihang Song et al. “Synthetic Datasets for Autonomous Driving: A Survey”. In: *IEEE Transactions on Intelligent Vehicles* 9.1 (Jan. 2024), pp. 1847–1864. ISSN: 2379-8904. DOI: 10.1109/TIV.2023.3331024.

[SSZ17] Jake Snell, Kevin Swersky, and Richard Zemel. “Prototypical Networks for Few-shot Learning”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017, pp. 4077–4087. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/cb8da6767461f2812ae4290eac7cbc42-Paper.pdf.

[Sta+23] Roman Stank et al. “Real-Time Wheel Detection and Rim Classification in Automotive Production”. In: *IEEE International Conference on Image Processing (ICIP)*. Kuala Lumpur, Malaysia: IEEE, Oct. 8, 2023, pp. 1410–1414. ISBN: 978-1-7281-9835-4. DOI: 10.1109/ICIP49359.2023.10223161.

[Sun+21] Bo Sun et al. “FSCE: Few-Shot Object Detection via Contrastive Proposal Encoding”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. virtual: IEEE, 2021, pp. 7352–7362. DOI: 10.1109/CVPR46437.2021.00727.

[SZ23] Sarvesh Sundaram and Abe Zeid. “Artificial Intelligence-Based Smart Quality Inspection for Manufacturing”. In: *Micromachines* 14.3 (Feb. 27, 2023), p. 570. ISSN: 2072-666X. DOI: 10.3390/mi14030570.

[Sze+16] Christian Szegedy et al. “Rethinking the Inception Architecture for Computer Vision”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA: IEEE, June 2016, pp. 2818–2826. ISBN: 978-1-4673-8851-1. DOI: 10.1109/CVPR.2016.308.

[Tan+21] Pengzhou Tang et al. “Image Dataset Creation and Networks Improvement Method Based on CAD Model and Edge Operator for Object Detection in the Manufacturing Industry”. In: *Machine Vision and Applications* 32.5 (Sept. 2021), p. 111. ISSN: 0932-8092. DOI: 10.1007/s00138-021-01237-y.

[Tao+22] Xian Tao et al. “Deep Learning for Unsupervised Anomaly Localization in Industrial Images: A Survey”. In: *IEEE Transactions on Instrumentation and Measurement* 71 (2022), pp. 1–21. ISSN: 0018-9456. DOI: 10.1109/TIM.2022.3196436.

[TC23] Juan Terven and Diana Cordova-Esparza. “A Comprehensive Review of YOLO Architectures in Computer Vision: From Yolov1 to Yolov8 and YOLO-NAS”. In: *Machine Learning and Knowledge Extraction* 5.4 (Nov. 20, 2023), pp. 1680–1716. ISSN: 2504-4990. DOI: 10.3390/make5040083.

[TL19] Mingxing Tan and Quoc V. Le. “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks”. In: *Proceedings of the 36th International Conference on Machine Learning (ICML)*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. Long Beach, California, USA: PMLR, 2019, pp. 6105–6114. URL: http://proceedings.mlr.press/v97/tan19a.html (visited on 05/26/2025).

[TV17] Antti Tarvainen and Harri Valpola. “Mean Teachers Are Better Role Models: Weight-averaged Consistency Targets Improve Semi-Supervised Deep Learning Results”. In: *5th International Conference on Learning Representations (ICLR)*. Toulon, France, Apr. 25, 2017. URL: https://openreview.net/forum?id=ry8u21rtl (visited on 05/26/2025).

[UYY21] Furkan Ulger, Seniha Esen Yuksel, and Atila Yilmaz. “Anomaly Detection for Solder Joints Using β-VAE”. In: *IEEE Transactions on Components, Packaging and Manufacturing Technology* 11.12 (Dec. 2021), pp. 2214–2221. ISSN: 2156-3950. DOI: 10.1109/TCPMT.2021.3121265.

[Wan+18] Jinjiang Wang et al. “Deep Learning for Smart Manufacturing: Methods and Applications”. In: *Journal of Manufacturing Systems* 48 (July 2018), pp. 144–156. ISSN: 0278-6125. DOI: 10.1016/j.jmsy.2018.01.003.

[Wan+20] Xin Wang et al. “Frustratingly Simple Few-Shot Object Detection”. In: *Proceedings of the 37th International Conference on Machine Learning (ICML)*. Vol. 119. Proceedings of Machine Learning Research. Virtual: PMLR, 2020, pp. 9919–9928. DOI: 10.5555/3524938.3525858.

[Wan+23] Haoyu Wang et al. “Synthetic Datasets for Rebar Instance Segmentation Using Mask R-CNN”. In: *Buildings* 13.3 (Feb. 22, 2023), p. 585. ISSN: 2075-5309. DOI: 10.3390/buildings13030585.

[Xie+17] Saining Xie et al. “Aggregated Residual Transformations for Deep Neural Networks”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI, USA, IEEE, 2017, pp. 5987–5995. DOI: 10.1109/CVPR.2017.634.

[Yan+19] Jing Yang et al. “Real-Time Tiny Part Defect Detection System in Manufacturing Using Deep Learning”. In: *IEEE Access* 7 (2019), pp. 89278–89291. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2019.2925561.

[Yan+20] Jing Yang et al. “Using Deep Learning to Detect Defects in Manufacturing: A Comprehensive Survey and Current Challenges”. In: *Materials* 13.24 (Dec. 16, 2020), p. 5755. ISSN: 1996-1944. DOI: 10.3390/ma13245755.

[YH22] Xinyi Yu and Yuanfu He. “PCB Defect Detection Based on GAN Data Generation with Self-attentive Mechanism”. In: *2nd International Conference on Frontiers of Electronics, Information and Computation Technologies (ICFEICT)*. Wuhan, China: IEEE, Aug. 2022, pp. 55–60. ISBN: 978-1-6654-5476-6. DOI: 10.1109/ICFEICT57213.2022.00018.

[Zha+18] Hongyi Zhang et al. “Mixup: Beyond Empirical Risk Minimization”. In: *International Conference on Learning Representations (ICLR)*. 2018. URL: https://openreview.net/forum?id=r1Ddp1-Rb.

[Zha+21a] Dingwen Zhang et al. “Weakly Supervised Object Localization and Detection: A Survey”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021), pp. 1–1. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2021.3074313.

[Zha+21b] Gang Zhao et al. “A Mask R-CNN Based Method for Inspecting Cable Brackets in Aircraft”. In: *Chinese Journal of Aeronautics* 34.12 (Dec. 2021), pp. 214–226. ISSN: 1000-9361. DOI: 10.1016/j.cja.2020.09.024.

[Zha+24a] Caicai Zhang et al. “On Efficient Expanding Training Datasets of Breast Tumor Ultrasound Segmentation Model”. In: *Computers in Biology and Medicine* 183 (Dec. 2024), p. 109274. ISSN: 0010-4825. DOI: 10.1016/j.compbiomed.2024.109274.

[Zha+24b] Fubao Zhang et al. “Real-Time Defect Detection of Saw Chains on Automatic Assembly Lines Based on Residual Networks and Knowledge Coding”. In: *Engineering Applications of Artificial Intelligence* 128 (Feb. 2024), p. 107507. ISSN: 0952-1976. DOI: 10.1016/j.engappai.2023.107507.