

**SANTA CATARINA STATE UNIVERSITY - UDESC
CENTER FOR TECHNOLOGICAL SCIENCES - CCT
MASTER'S IN APPLIED COMPUTING**

WELINTON TRENTIN

**AUTOMATED INFORMATION EXTRACTION FROM ENGINEERING
DRAWINGS USING COMPUTER VISION AND MULTIMODAL LARGE
LANGUAGE MODEL**

JOINVILLE

2026

WELINTON TRENTIN

**AUTOMATED INFORMATION EXTRACTION FROM ENGINEERING
DRAWINGS USING COMPUTER VISION AND MULTIMODAL LARGE
LANGUAGE MODEL**

Dissertation submitted to the Graduate Program in Applied Computing of the Center for Technological Sciences of the Santa Catarina State University in partial fulfillment of the requirements for the degree of Master in Applied Computing.

Advisor: Prof. Dr. Fabiano Baldo

JOINVILLE

2026

**Ficha catalográfica elaborada pelo programa de geração automática da
Biblioteca Universitária Udesc,
com os dados fornecidos pelo(a) autor(a)**

Trentin, Welinton
AUTOMATED INFORMATION EXTRACTION FROM
ENGINEERING DRAWINGS USING COMPUTER VISION
AND MULTIMODAL LARGE LANGUAGE MODE / Welinton
Trentin. -- 2026.
108 p.

Orientador: Fabiano Baldo
Dissertação (mestrado) -- Universidade do Estado de
Santa Catarina, Centro de Ciências Tecnológicas, Programa
de Pós-Graduação em Computação Aplicada, Joinville, 2026.

1. Convolutional Neural Networks. 2. Multimodal Large
Language Models. 3. Engineering Drawings. 4. Information
Extraction. 5. Industrial Automation. I. Baldo, Fabiano. II.
Universidade do Estado de Santa Catarina, Centro de
Ciências Tecnológicas, Programa de Pós-Graduação em
Computação Aplicada. III. Título.

WELINTON TRENTIN

**AUTOMATED INFORMATION EXTRACTION FROM ENGINEERING DRAWINGS
USING COMPUTER VISION AND MULTIMODAL LARGE LANGUAGE MODEL**

This dissertation was deemed suitable for obtaining the degree of **Master in Applied Computing**, concentration area in “Computer Science”, and approved in its final form by the Graduate Program in Applied Computing of the Center for Technological Sciences of the Santa Catarina State University.

Examination Board:

Prof. Dr. Fabiano Baldo
CCT/UDESC
(Advisor)

Dr. Alexandre Gonçalves Silva
INE/UFSC

Dr. Yuri Kaszubowski Lopes
CCT/UDESC

Joinville, July 08, 2026

To my wife Francine, my children José Bento and Maria Guadalupe, to my parents Clovis and Rosani for their encouragement in my studies, and to my brother Vinicius for being an inspiration.

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my wife, Francine, for her unwavering support, encouragement, and understanding throughout this journey. Her patience and confidence in me were invaluable to the completion of this work.

I would like to sincerely thank my advisor, Prof. Dr. Fabiano Baldo, for his guidance, insightful discussions, and continuous support throughout the development of this research. His expertise and encouragement were fundamental to the progress of this project.

I am also grateful to WEG for providing the opportunity, resources, and time necessary to carry out this research and develop the proposed solution.

I would also like to express my sincere gratitude to the Santa Catarina Research and Innovation Foundation (FAPESC) for its financial support, which made this research possible and contributed significantly to the successful completion of this dissertation.

Finally, I would like to thank UDESC for the opportunity to pursue this Master's degree and for providing an environment that fostered learning, research, and professional growth.

*“All to Jesus through Mary, all to Mary for
Jesus.”*

ABSTRACT

Maintaining consistency between industrial production databases and the information contained in engineering drawings remains a significant challenge in customized manufacturing environments. Since production and quality verification activities are commonly performed using engineering drawings in PDF format rather than native CAD files, the validation process depends on unstructured documents that lack semantic information and are frequently verified manually. This software-dependent workflow is time-consuming, prone to human error, and difficult to scale. To address these limitations, this work proposes a multimodal framework capable of extracting and validating dimensional information directly from engineering drawings, independently of the original CAD platform. The proposed approach combines hierarchical Convolutional Neural Networks (CNNs) for visual localization and segmentation with Multimodal Large Language Models (MLLMs) for joint visual-textual reasoning, enabling the contextual interpretation of dimensions and tolerances. To improve reliability, a sequential dual-verification strategy is employed, where the semantic interpretation produced by the MLLM is validated through targeted Optical Character Recognition (OCR), reducing hallucinations and increasing extraction robustness. The framework was evaluated using 77 real industrial engineering drawings comprising more than 15,400 individual verification tasks. Experimental results demonstrate consistent performance, achieving Precision, Recall, and F1-Score values above 80%, confirming the proposed framework as an effective and scalable solution for automating engineering drawing validation in Industry 4.0 environments.

Keywords: Computer Vision, Convolutional Neural Networks, Multimodal Large Language Models, Engineering Drawings, Information Extraction, Industrial Automation.

LIST OF FIGURES

Figure 1 – Example of a complex industrial machine handled in the study case.	16
Figure 2 – Example of a technical documentation	16
Figure 3 – YOLOv8 Detailed Structural Pipeline	25
Figure 4 – YOLOv1 architecture	27
Figure 5 – K-Fold Cross-Validation	34
Figure 6 – Schematic of the proposed multi-stage automated workflow.	45
Figure 7 – Example of the source engineering document ingested by the system.	46
Figure 8 – Application of Model 1 to identify all machine view regions.	46
Figure 9 – Application of Model 2 to classify the detected view as “Side View”. .	47
Figure 10 – MLLM input image	48
Figure 12 – Dimension–tolerance separation	49
Figure 11 – Application of Model 3 to detect all dimension bounding boxes. . . .	49
Figure 13 – Verification overlay	51
Figure 14 – Verification output	52
Figure 15 – Gamification tool interface	55
Figure 16 – Pareto distribution of classes	58
Figure 17 – View classification (Step II)	59
Figure 18 – Bounding box localization (Step III)	59
Figure 19 – Value–tolerance decomposition (Step IV)	60
Figure 20 – Results of the 40-trial hyperparameter optimization study.	62
Figure 21 – Hyperparameter importance	63
Figure 22 – SHAP summary plot	64
Figure 23 – Training and Validation Box Losses Across 5 K-Folds - Model 1 . . .	66
Figure 24 – Training and Validation Classification Losses Across 5 K-Folds - Model 1	67
Figure 25 – Training and Validation DFL Losses Across 5 K-Folds - Model 1 . . .	67
Figure 26 – Model 1 validation metrics	68
Figure 27 – Training and Validation Box Losses Across 5 K-Folds - Model 2 . . .	70
Figure 28 – Training and Validation Classification Losses Across 5 K-Folds - Model 2	70
Figure 29 – Training and Validation DFL Losses Across 5 K-Folds - Model 2 . . .	70
Figure 30 – Model 2 performance (validation)	71
Figure 31 – Training and Validation Box Losses Across 5 K-Folds - Model 3 . . .	73
Figure 32 – Training and Validation Classification Losses Across 5 K-Folds - Model 3	73
Figure 33 – Training and Validation DFL Losses Across 5 K-Folds - Model 3 . . .	73
Figure 34 – Model 3 performance (validation)	74
Figure 35 – Training and Validation Box Losses Across 5 K-Folds - Model 4 . . .	76
Figure 36 – Training and Validation Classification Losses Across 5 K-Folds - Model 4	76

Figure 37 – Training and Validation DFL Losses Across 5 K-Folds - Model 4 . . .	76
Figure 38 – Model 4 performance (validation)	77
Figure 39 – Performance heatmaps (measures)	83
Figure 40 – Performance heatmaps (dimensions)	84
Figure 41 – Example in D39 - Pipeline failure caused by the crossing line. . . .	85
Figure 42 – Pipeline success after removing the crossing line.	86
Figure 43 – Example D25 - Pipeline success without removing the crossing line.	87
Figure 44 – MLLM reasoning with occlusion	88
Figure 45 – OCR limitation (single-digit)	89
Figure 46 – Verification success (D36, T1)	89
Figure 47 – Verification failure (D36, T2)	90
Figure 48 – Example in D32 - Data conflict	91
Figure 49 – Example in D56 - A detection failure.	92
Figure 50 – MLLM input (D36)	93
Figure 51 – Pipeline output (D36)	93
Figure 52 – MLLM errors (D5)	95

LIST OF TABLES

Table 1 – Summary of Research Classification.	20
Table 2 – Mapping of the reviewed articles and their corresponding references.	39
Table 3 – Cross-analysis of the selected studies and their thematic relationships.	40
Table 4 – Optuna hyperparameter search space.	61
Table 5 – Optimal hyperparameters identified by the best-performing trial.	62
Table 6 – K-Fold Cross-Validation Summary - Model 1 (Global View Detector).	65
Table 7 – Label distribution for orthographic machine view classes.	69
Table 8 – K-Fold Cross-Validation Summary - Model 2 (Orientation Classifier).	69
Table 9 – K-Fold Cross-Validation Summary - Model 3 (Dimension Block Locator).	72
Table 10 – K-Fold Cross-Validation Summary - Model 4 (Tolerance Classifier).	75
Table 11 – Performance Metrics for Measures and Dimensions	81

LIST OF ABBREVIATIONS AND ACRONYMS

AI	Artificial Intelligence
ANN	Artificial Neural Networks
BCE	Binary Cross Entropy
CAD	Computer-Aided Design
CIoU	Complete Intersection over Union
CV	Computer Vision
CNN	Convolutional Neural Networks
DFL	Distribution Focal Loss
ERP	Enterprise Resource Planning
GFL	Generalized Focal Loss
mAP	Mean Average Precision
MLLM	Multimodal Large Language Model
NLP	Natural Language Processing
OCR	Optical Character Recognition
OD	Object Detection
PDF	Portable Document Format
VF	Front View (<i>Vista Frontal</i>)
VL	Side View (<i>Vista Lateral</i>)
VS	Top View (<i>Vista Superior</i>)
YOLO	You Only Look Once

TABLE OF CONTENTS

1	INTRODUCTION	15
1.1	Motivation	17
1.2	Objective	18
1.2.1	Specific Objectives	18
1.3	Research Methodology	19
1.3.1	Methodological Framework	19
1.3.1.1	Scientific Areas and Interdisciplinarity	19
1.3.1.2	Nature and Paradigm	19
1.3.1.3	Methodological Approach	20
1.3.1.4	Research Procedures	20
1.4	Structure of this work	21
2	BACKGROUND	23
2.1	Theoretical Foundation	23
2.1.1	Technical Documentation in Engineering	23
2.1.2	Computer Vision: Enabling Machines to See Drawings	23
2.1.3	Convolutional Neural Networks (CNNs)	24
2.1.4	Object Detection Architectures	24
2.1.4.1	Two-Stage Detectors: The R-CNN Family	24
2.1.4.2	One-Stage Detectors and the Single-Shot Paradigm	25
2.1.4.3	YOLO: Classical Foundations and Structural Evolution	26
2.1.4.4	YOLOv8: Architectural Advancements in Technical Layouts	28
2.1.5	Hyperparameter Tuning	30
2.1.5.1	Key CNN Hyperparameters for Optimization	30
2.1.5.2	Hyperparameter Optimization with Optuna	31
2.1.6	K-Fold Cross-Validation	33
2.1.7	Optical Character Recognition (OCR)	34
2.1.8	Natural Language Processing	35
2.1.8.1	Multimodal Large Language Models (MLLMs)	35
2.1.9	Evaluation Metrics	36
2.2	Related Works	37
2.2.1	Review Methodology	37
2.2.2	Synthesis of Findings	38
2.2.3	Selected Studies Analysis	38
2.2.3.1	Thematic Discussion and Evolution	40

2.2.4	Reported Challenges	41
2.2.5	Positioning and Contributions of this Work	43
3	PROPOSED APPROACH	44
3.1	Conceptual Framework	44
3.2	Proposed Workflow in Details	45
3.3	Chapter Considerations	52
4	DEVELOPMENT METHODOLOGY	53
4.1	Database Preparation	54
4.1.1	Dataset Selection and Temporal Scope	54
4.1.2	Collaborative Labeling and Gamification	54
4.1.2.1	User Onboarding and Operational Modes	55
4.1.2.2	Core Labeling Loop and Ergonomic Controls	55
4.1.2.3	Competitive Mechanics and Quality Assurance	56
4.1.2.4	Quality Assurance Labeling	56
4.1.3	Hierarchical Labeling Workflow	57
4.1.3.1	Step I: Electric Machine View Counting	57
4.1.3.2	Step II: Spatial Orientation Labeling	58
4.1.3.3	Step III: Dimension Block Localization	58
4.1.3.4	Step IV: Fine-Grained Dimension and Tolerance Decomposition	59
4.2	Models Training	60
4.2.1	Hyperparameter Optimization Strategy	60
4.2.2	Optimization Setup and Search Space	61
4.2.3	Tuning Results and Sensitivity Analysis	61
4.2.3.1	Parameter Importance Analysis	62
4.2.4	K-Fold Cross-Validation	64
4.2.5	Model 1: Machine View Detection	65
4.2.5.1	Training and Cross-Validation Analysis	65
4.2.5.2	Temporal Out-of-Sample Validation	67
4.2.6	Model 2: Machine View Type Classification	68
4.2.6.1	Training and Cross-Validation Analysis	68
4.2.6.2	Temporal Out-of-Sample Evaluation	71
4.2.7	Model 3: Machine Dimension Detection	71
4.2.7.1	Training and Cross-Validation Analysis	72
4.2.7.2	Temporal Out-of-Sample Evaluation	74
4.2.8	Model 4: Dimension and Tolerance Classification	74
4.2.8.1	Training and Cross-Validation Analysis	75
4.2.8.2	Temporal Out-of-Sample Evaluation	77

4.3	Multimodal Reasoning and Semantic Verification	77
4.3.1	Prompt Engineering and Instruction Design	78
4.3.2	Multimodal Input Pipeline	78
4.4	Evaluation Pipeline	79
4.4.1	Dataset Composition	79
4.4.2	Evaluation Tasks: Measures vs. Dimensions	79
4.5	Chapter Considerations	80
5	RESULTS	81
5.1	End-to-End System Pipeline Results	81
5.2	Analysis of Challenges and Edge Cases	84
5.2.1	Challenge: Impact of Geometric Interference on Data Extraction	85
5.2.2	Challenge: Ambiguity in Single-Digit Character Recognition . . .	88
5.2.3	Challenge: Performance Variability and Result Oscillation	89
5.2.4	Challenge: Data Conflict and False Positives	90
5.2.5	Challenge: Missed Detection from Incorrect Cropping	91
5.2.6	Challenge: Contextual Recovery in Non-Standard Drawing Layouts	92
5.2.7	Challenge: Semantic Misalignment and MLLM Hallucinations . .	93
5.3	Chapter Considerations	95
6	CONCLUSION	97
6.1	Future Work	97
	REFERENCES	99
A	APPENDIX	104
A.1	MLLM Prompt Generation Function	104
A.2	Analyzed Measures and Definitions	106

1 INTRODUCTION

The transition towards Industry 4.0 has pressured manufacturing sectors to achieve unprecedented levels of data synchronization across the entire production chain. However, as noted by Kagermann et al. (2013), the digitalization of legacy processes remains a significant challenge. A recurring manifestation of this issue is the discrepancy between information recorded in the Production System database and the technical specifications contained in engineering drawings. These inconsistencies between document-based data and system records can compromise product integrity, trigger delivery delays, and generate substantial rework (ZHONG et al., 2017).

In the specific industrial context of this research—a large-scale manufacturer of customized equipment—these inconsistencies present a critical bottleneck. Currently, at this organization, the verification of such data is performed manually. This requires highly skilled professionals to execute repetitive tasks, which, as highlighted by Reason (1990), are inherently prone to human error due to cognitive fatigue and represent an inefficient allocation of intellectual capital.

To illustrate the complexity of this task, Figure 1 presents one of the main one-of-a-kind machines produced by the company. The technical documentation for such equipment involves dense 2D drawings (Figure 2) with dimensions and tolerances that must match the digital Production System.

A fundamental paradox exists in this workflow: while the design phase utilizes sophisticated Computer-Aided Design (CAD) ecosystems, the subsequent verification, validation and production stages typically rely on flattened, unstructured documentation such as PDF files. These documents lack embedded metadata and do not allow for the direct use of vector source files (e.g., .dwg) in the verification process. Consequently, the transition from a high-fidelity digital model (CAD files) to a static document (PDF files) results in a loss of semantic structure, forcing the validation process to rely on human visual interpretation rather than automated data cross-referencing.

Furthermore, relying on CAD-native data would significantly limit the applicability of the proposed solution. In industrial environments, engineering documentation may originate from different CAD platforms, be archived as scanned documents, or even be available only as rasterized PDF files after years of revisions and supplier exchanges. Therefore, the proposed framework is intentionally designed to operate directly on engineering documents, regardless of their origin or whether the original CAD model is available. This software-independent approach increases its robustness, portability, and applicability to heterogeneous industrial environments.



Figure 1 – Example of a complex industrial machine handled in the study case.

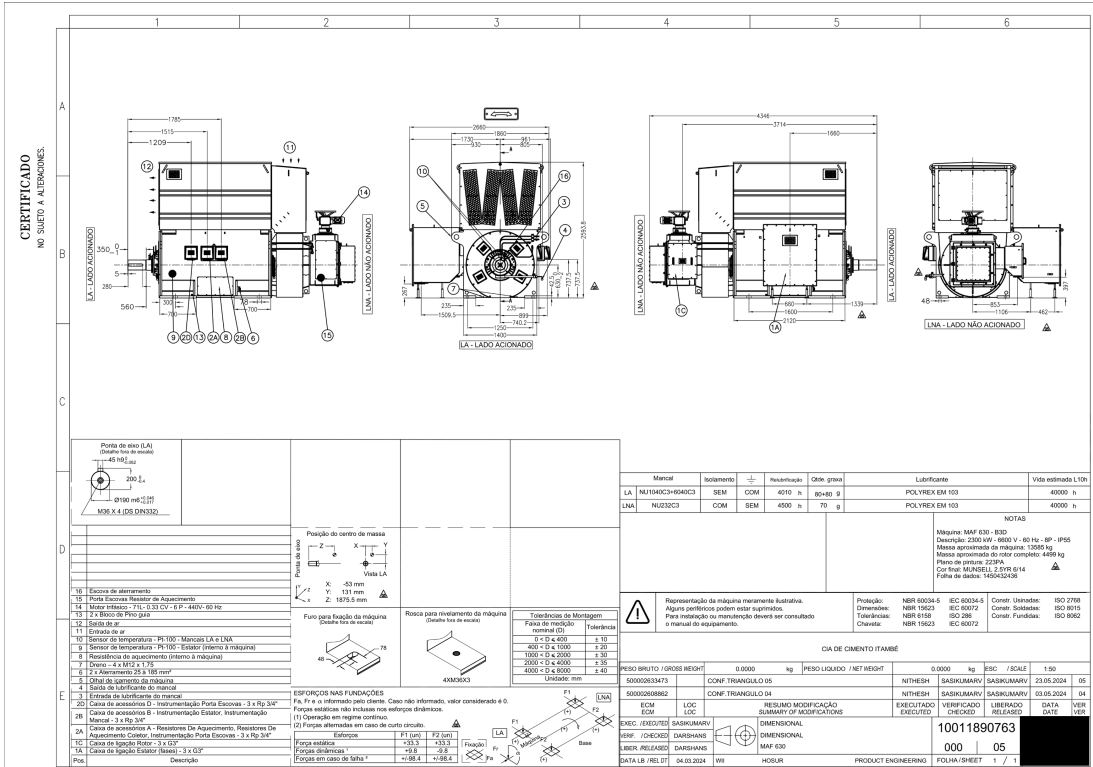


Figure 2 – Example of a technical documentation

In this context, this work proposes a framework that combines advanced Computer Vision (CV) techniques, specifically Convolutional Neural Networks (CNNs), with Multi-Modal Large Language Models (MLLMs) to automate the extraction and semantic interpretation of information from engineering drawings. MLLMs are employed because the interpretation of technical drawings requires the joint understanding of visual and textual information, where dimensions, tolerances, symbols, and annotations must be analyzed together with their spatial relationships. This multimodal reasoning enables contextual identification and validation of extracted information before comparison with the Production System database.

The scientific and practical relevance of this study lies in its ability to promote greater document reliability and operational agility, mitigating possible production errors due to out-of-date information contained in the PDF drawings in comparison of those presented in the Production System database. Therefore, it directly contributes to the digitalization and automation of industrial processes in alignment with the principles of Industry 4.0.

1.1 MOTIVATION

A critical challenge arises in industries focused on customized (one-of-a-kind) manufacturing, such as the production of large-scale electric motors, where ad-hoc modifications are frequently made to accommodate manufacturing constraints and new customers' refinements. Often, these alterations are annotated directly on the physical or digital drawing but are not immediately propagated to the organization's central Production System database. Such discrepancies create a "data silo" effect, leading to a divergence between the documented drawings and actual states of the project in the Production System, which may result in the fabrication of incompatible subcomponents or critical assembly errors (ZHONG et al., 2017).

Currently, this company tries to mitigate these risks by manually auditing each component's measures contained in the drawing documents against the Production System at each production stage. This is a time-consuming process, inherently prone to human cognitive fatigue and errors (REASON, 1990).

One intuitive way to automatically solve this problem is by applying an Optical Character Recognition (OCR) technique which can recognise the measures in the engineering drawings and then compare them to the values presented in the Production System. However, while generic OCR models demonstrate efficacy in simple environments, such as license plate recognition (SMITH, 2007), they exhibit significant limitations when applied to the interpretation of complex 2D engineering drawings. These documents are characterized by a high density of information, where alphanumeric

characters are deeply intertwined with geometric primitives (lines, arrows, and symbols). In manufacturing, the precision of these annotations is vital for production integrity (WANG; ZHOU et al., 2020).

Therefore, addressing this problem requires the adoption of a more comprehensive approach that integrates finer-grained drawing segmentation and classification with an additional layer of semantic interpretation of the extracted information, enabling the identification of the specific measurement to which each element refers. In this regard, the integration of CV techniques, such as CNN, with the reasoning capabilities of MLLMs constitutes a relevant path toward automation. Such an approach enables not only the localization and extraction of textual information, but also the semantic inference of measurement meaning based on its spatial context within the 2D design environment. As a result, these techniques contribute to reducing inconsistencies between drawings and systems, with positive implications for operational efficiency.

1.2 OBJECTIVE

The primary objective of this work is to develop a multimodal framework capable of accurately extracting and validating dimensional information from engineering drawings. The proposed framework integrates Convolutional Neural Networks (CNNs) for precise visual localization and segmentation with Multimodal Large Language Models (MLLMs) for semantic interpretation and contextual validation against the Industrial Production System database.

1.2.1 Specific Objectives

To achieve the primary objective, the following specific goals were defined:

- **Investigate the state-of-the-art** in Computer Vision and Natural Language Processing, focusing on how modern models have evolved from simple text recognition (OCR) to contextual interpretation of technical documents.
- **Construct and curate a proprietary dataset** of real-world industrial engineering drawings, implementing a rigorous annotation protocol to establish a reliable ground truth for training and benchmarking.
- **Develop and optimize Object Detection models** based on CNN architectures (such as YOLO) to localize critical regions of interest, effectively isolating dimension blocks and technical specifications from complex geometric noise.
- **Implement a Reasoning and Validation layer** using MLLMs and advanced Prompt Engineering to contextually verify extracted data against predefined engineering rules and system records.

- **Design a multimodal extraction pipeline** that integrates visual feature extraction and text recognition (CNN + MLLM), preserving the spatial relationships required for the accurate interpretation of dimensions and tolerances.
- **Evaluate the proposed pipeline** through a comprehensive performance analysis using Mean Average Precision (mAP) for object detection and Precision, Recall, and F1-Score for information extraction in real-world industrial scenarios.

1.3 RESEARCH METHODOLOGY

This section details the methodological approach adopted to address the research problem and achieve the aimed objective. It provides a comprehensive description of the research classification and the methodological procedure adopted to develop the proposed framework.

1.3.1 Methodological Framework

This research is classified and characterized according to its scientific areas, nature, paradigm, and methodological criteria. The following subsections detail these classifications.

1.3.1.1 Scientific Areas and Interdisciplinarity

This study is interdisciplinary, bridging the gap between **Computer Science** and **Engineering** (CAPES, 2020).

- **Exact and Earth Sciences:** Specifically within **Computer Science (1.03.00.00-7)**, focusing on **Methodology and Computing Techniques (1.03.03.00-6)**.
- **Engineering:** Specifically within **Mechanical Engineering (3.05.00.00-1)**, with an emphasis on **Machine Design (3.05.04.00-7)**.

This classification reflects the necessary integration of CV and NLP techniques applied to the domain of technical engineering drawings.

1.3.1.2 Nature and Paradigm

The research is classified as **Applied Science** (WAZLAWICK, 2010), as it aims to develop and test practical solutions for automating information extraction and validation in real-world contexts. It adopts a **Technocratic Paradigm**, prioritizing validation through experimentation and the measurement of results in practical scenarios (EDEN, 2007). Regarding maturity, it follows the **Applied nature (Style 3)** (WAZLAWICK, 2010), exploring and consolidating existing technologies (CV and NLP) to address the specific

research question regarding the automation of information extraction and validation from 2D drawings.

1.3.1.3 Methodological Approach

The methodological characteristics of the research are defined as follows:

- **General Objective:** The research is **Descriptive**, aiming to detail the process, variables, and results of applying CV and NLP techniques (GIL, 2010).
- **Logical Reasoning:** It primarily employs **Inductive reasoning** to generate generalizations from specific cases, while also using **Deductive reasoning** to validate hypotheses regarding technical parametrization (MARCONI; LAKATOS, 2005; JR.; PEREIRA; FILHO, 2007; PRODANOV; FREITAS, 2013).
- **Nature of Variables:** The approach is **Quantitative**. It analyzes quantitative metrics like Precision, Recall, F1-score, and so forth (GIL, 2010).
- **Data Collection Procedures:** The collection is **direct**, featuring a **quasi-experiment** conducted in an industrial engineering environment using technical documents. This is complemented by **prototyping experiments** (MARCONI; LAKATOS, 2005).
- **Variable Control:** The research is **Experimental** with partial control of variables because the real-world industrial conditions may influence results (GIL, 2010).

Table 1 summarizes the methodological approach.

Table 1 – Summary of Research Classification.

Classification Criteria	Definition
Scientific Area	Computer Science & Mechanical Engineering
Nature	Applied Science
Paradigm	Technocratic
Maturity	Applied (Style 3)
Objective	Descriptive
Reasoning	Inductive and Deductive
Approach	Quantitative
Procedure	Quasi-experiment, Prototyping

1.3.1.4 Research Procedures

The development of this work followed a structured methodological workflow divided into eight distinct stages. These procedures are detailed below:

1. **Bibliographic Review:** A systematic review of the literature was conducted to identify state-of-the-art techniques in CV, NLP, and their applications in technical document analysis. This stage focused on the evolution from traditional OCR/NLP pipelines to modern convolutional and multimodal reasoning.
2. **Dataset Acquisition:** A proprietary database of technical drawings was collected from the industrial environment. This dataset was curated to capture the real-world variability, complexity, and specificities of engineering documents used in customized manufacturing.
3. **Data Annotation:** Raw images were manually annotated to identify critical Regions of Interest (RoIs), such as machinery faces, dimension blocks, tolerances, and technical notes. This process established the “ground truth” necessary for training and benchmarking supervised learning models.
4. **Data Validation:** A rigorous verification step followed the annotation process to ensure label consistency and quality. This stage was crucial to minimize noise and bias that could degrade model convergence during training.
5. **Model Training:** The computer vision models (specifically CNNs) were trained using part of the annotated dataset. This stage involved rigorous hyperparameter optimization to maximize model performance without overfitting.
6. **Prompt Engineering:** Specialized prompts were designed to guide interactions with the MLLM, enabling the semantic contextualization of information extracted from the drawing documents and allowing the MLLM to interpret it and construct answers. This step was crucial for standardizing units and correcting OCR inconsistencies.
7. **Framework Proposal and Integration:** The trained CV models and optimized multimodal strategies were integrated into a cohesive **pipeline**. This architecture defines the end-to-end flow for automated information extraction and cross-verification with the Production System database.
8. **Results Analysis:** A comprehensive evaluation of the workflow was performed using quantitative metrics, such as Mean Average Precision, Precision, Recall, and F1-Score, followed by qualitative analysis of particular cases.

1.4 STRUCTURE OF THIS WORK

This document is organized into distinct chapters, structured as follows:

Chapter 2 presents the Background and Related Works, establishing the theoretical foundation that supports this research and highlighting relevant studies in the literature that address similar problems.

Chapter 3 describes the proposed approach, explaining the reasoning behind the solution and detailing the final architecture of the data extraction and validation workflow.

Chapter 4 presents the framework developed to implement the automated extraction and verification of dimensional data from engineering drawings. It details the operational workflow, ranging from the optimization of training hyperparameters to the systematic evaluation of the four specialized models through K-fold cross-validation.

Chapter 5 details the assessment performed and the experimental findings, providing a comprehensive discussion regarding the performance of the integrated information extraction workflow. It presents a quantitative evaluation of the end-to-end pipeline, highlighting metrics such as Precision, Recall, and F1-Score. Finally, the chapter concludes with a critical analysis of the primary challenges encountered, specifically focusing on the visual complexity of engineering drawings and the strategies developed to maintain data integrity.

Finally, Chapter 6 summarizes the conclusions, discussing the main contributions achieved at this stage and providing insights for future works.

2 BACKGROUND

This chapter establishes the theoretical and empirical foundations of this work. It is divided into two main parts: Section 2.1 presents the fundamental concepts required to understand the technologies utilized in this work, ranging from Computer Vision principles to Multimodal Large Language Models. Section 2.2 reviews the related works, discussing state-of-the-art approaches and identifying the technological gaps that this research aims to address.

2.1 THEORETICAL FOUNDATION

This section presents the fundamental concepts required to understand the technologies and methodologies involved in extracting information from engineering technical drawings. The content is structured from general principles to specific approaches, providing the necessary context for the architectural decisions taken in this work and detailed in later chapters.

2.1.1 Technical Documentation in Engineering

Technical drawings are essential tools that function as a universal graphic language in industrial environments, bridging communication between design and manufacturing stages. Their primary purposes are visualization, communication, and documentation. To ensure these drawings are interpreted correctly by all stakeholders, their standardization is governed by technical norms from bodies like American National Standards Institute (ANSI), American Society of Mechanical Engineers (ASME), and International Organization for Standardization (ISO). Drawings evolve through the design process: they begin as technical sketches during the ideation phase and are refined into precise production or working drawings in the final stages. These final drawings provide all necessary details for manufacturing and form a critical part of the overall product definition (GIESECKE et al., 2014). The inherent complexity and information density of these documents, however, make manual data extraction a time-consuming and error-prone process. Automating this extraction is where computational methods become indispensable.

2.1.2 Computer Vision: Enabling Machines to See Drawings

The automation of technical drawing interpretation fundamentally relies on the ability to computationally perceive and process visual data. This is the core domain of Computer Vision, a branch of Artificial Intelligence dedicated to enabling systems

to interpret and understand the visual world. Through the application of these techniques, static images can be transformed into structured formats suitable for automated analysis. Fundamental operations—such as edge detection, feature extraction, and image segmentation—serve as initial steps in deconstructing the graphical elements of a drawing, ranging from basic lines to complex component geometries (GONZALEZ; WOODS, 2018).

2.1.3 Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs), a specialized branch of Artificial Neural Networks (ANN), have become the cornerstone of modern image analysis. Unlike older methods that require manual feature engineering, CNNs automatically learn a hierarchy of features — from simple edges to complex geometric patterns — directly from the raw pixel data (O'SHEA; NASH, 2015).

A standard CNN architecture is composed of three main types of layers:

- **Convolutional Layers:** The core building blocks that apply learnable filters to extract spatial features.
- **Pooling Layers:** Used for dimensionality reduction, helping the model maintain translation invariance and reducing computational load.
- **Fully Connected Layers:** Typically located at the end of the network to perform the final classification or regression based on the features extracted by the previous layers.

2.1.4 Object Detection Architectures

Object detection extends simple image classification by identifying both “what” an object is and “where” it is spatially located within a matrix. In complex technical drawings, this computational task allows for the precise localization of components, symbols, and dimension lines regardless of scale, contrast variation, or orientation (ZOU et al., 2023). These modern architectures are generally categorized into two distinct paradigms:

2.1.4.1 Two-Stage Detectors: The R-CNN Family

Two-stage detectors, such as Faster R-CNN, approach the detection task sequentially, whereas the first stage utilizes a Region Proposal Network (RPN) to scan the image and propose potential regions of interest. The second stage then extracts features from these proposed zones, classifies the contents, and refines their bounding box coordinates. While highly accurate — especially for tiny, low-contrast components

in complex layouts — they require high computational intensity and lack the inference speed necessary for high-throughput pipelines due to their dual-pass architecture (REN et al., 2015).

2.1.4.2 One-Stage Detectors and the Single-Shot Paradigm

One-stage detectors, such as **YOLO (You Only Look Once)** and **SSD (Single Shot MultiBox Detector)**, treat object detection as a unified regression problem. By bypassing the region proposal phase entirely, these networks map raw input pixels directly to final bounding box coordinates and categorical class probabilities in a single forward propagation pass through the network (ZOU et al., 2023). This single-shot behavior drastically increases processing speed, making them highly suitable for industrial automation frameworks where operational efficiency is paramount.

The Single-Shot Mechanism Flow

The fundamental computational flow of a single-shot system ensures that the global image tensor is evaluated concurrently. As illustrated across the internal structural pipelines (Figure 3), the input image maps straight to the multi-scale predictions through four coordinated steps:

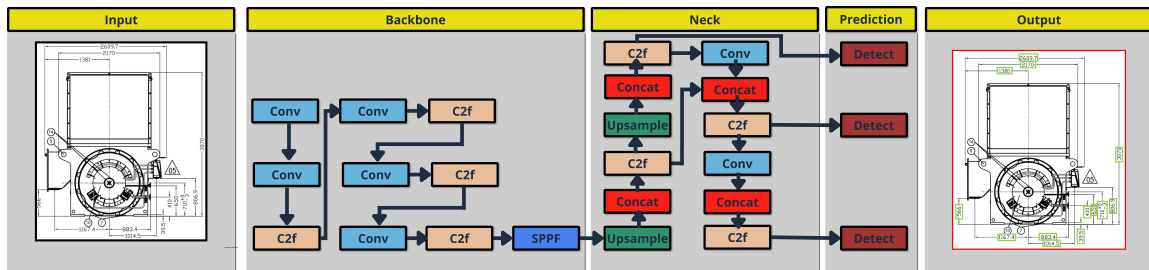


Figure 3 – Detailed structural pipeline of the YOLOv8 architecture, highlighting the C2f feature extraction flow in the Backbone, PANet multi-scale fusion in the Neck, and the decoupled detection branches in the Prediction stage (Adapted from Herfandi et al. (2023)).

1. **Unified Spatial Input:** The raw engineering document enters the **Input** layer. Rather than cropping or generating selective search patches, the network treats the global canvas as a single unified tensor, maintaining the spatial and contextual relationships between technical annotations and adjacent drawing lines (REDMON et al., 2016; ZOU et al., 2023).
2. **Single-Pass Feature Extraction (Backbone):** As the tensor flows linearly through the Backbone's convolutional (*Conv*) and Cross-stage Partial (*C2f*) layers, features are downsampled and aggregated concurrently (JOCHER; CHAURASIA; JING, 2023). High-level semantics and macro layout contexts are captured in a

single pass up to the Spatial Pyramid Pooling Fast (*SPPF*) block, optimizing the receptive field features while completely eliminating redundant backbone multi-scale evaluations (TERVEN; CORDOVA-ESPARZA, 2023).

3. **Concurrent Multi-Scale Fusion (Neck):** The multi-scale features enter the Neck, where top-down and bottom-up pyramids utilize *Upsample* and *Concat* blocks based on the Path Aggregation Network (PANet) topology (LIU et al., 2018). Instead of evaluating different drawing elements at separate operational cycles, the network maps both macro-features (machine views) and micro-features (tolerances) into a shared multi-scale feature map simultaneously to mitigate spatial degradation across dense layers (JOCHER; CHAURASIA; JING, 2023; TERVEN; CORDOVA-ESPARZA, 2023).
4. **Direct Regression (Prediction to Output):** The pipeline culminates in the decoupled Prediction layer (*Detect* heads). Because the architecture shifts to an anchor-free task paradigm (GE et al., 2021), it performs a direct mathematical regression from the fused multi-scale maps. In one simultaneous forward calculation, the network solves for:

$$\hat{Y} = f(X) \quad (2.1)$$

where X represents the global image matrix and $\hat{Y}$ is a structured tensor encoding bounding box boundaries (via Distribution Focal Loss) and categorical confidence matrices (via Binary Cross Entropy) for all objects at once (LI et al., 2020; JOCHER; CHAURASIA; JING, 2023).

Finally, a fast *Non-Maximum Suppression (NMS)* filter post-processes the tensor to suppress redundant overlapping bounding boxes, yielding the single-shot **Output** (REDMON et al., 2016). This streamlined computational flow explains why the cascade of four specialized models remains highly efficient for industrial automation; each individual network extracts its targets in a single forward pass, preventing processing bottlenecks.

2.1.4.3 YOLO: Classical Foundations and Structural Evolution

The original YOLO architecture (YOLOv1) revolutionized computer vision by redefining object detection. Rather than repurposing classifiers to scan images at multiple locations and scales, it proposed a unified approach by dividing the input image into an $S \times S$ grid (REDMON et al., 2016). Each individual grid cell is responsible for predicting B bounding boxes, their respective confidence scores, and C conditional class probabilities only if the center of an object falls within that specific cell's boundaries. This comprehensive spatial distribution allows the model to reason globally about the visual context and layout topology, significantly outperforming traditional sliding-window or

two-stage region-based frameworks in high-throughput real-time applications (REDMON et al., 2016; ZOU et al., 2023).

As illustrated in Figure 4, the foundational processing network consists of 24 convolutional layers followed by 2 fully connected (connected) layers. Heavily inspired by the GoogLeNet architecture for image classification, the network alternates standard spatial feature extraction with alternating layers to optimize feature abstraction. The pipeline can be decomposed into three primary operational blocks:

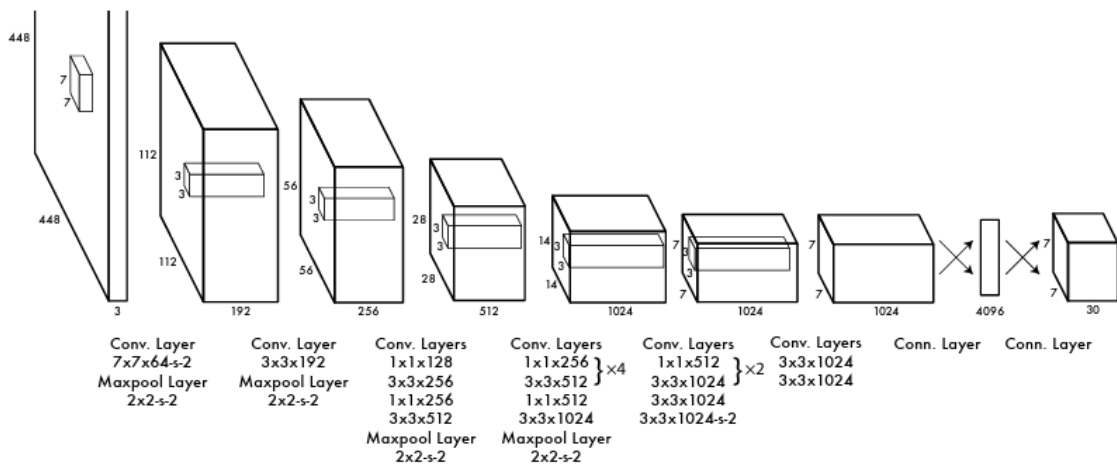


Figure 4 – Detailed schematic of the original YOLO architecture. The pipeline treats detection as a unified regression problem, processing a $448 \times 448 \times 3$ input tensor through a series of convolutional layers to map final predictions into a dense $7 \times 7 \times 30$ output tensor (REDMON et al., 2016).

- **Coarse Feature Extraction and Downsampling:** The pipeline ingests a standardized $448 \times 448 \times 3$ image tensor. The initial layers apply large receptive fields (such as the first $7 \times 7 \times 64$ convolution with a stride of 2) coupled with 2×2 Maxpool layers. This rapidly compresses the spatial dimensions ($448 \rightarrow 112 \rightarrow 56$) while capturing fundamental geometric primitives like edges, high-contrast borders, and basic shapes.
- **Dimensional Reduction Modules (1×1 Convolutions):** To manage computational complexity and prevent catastrophic memory overhead, the intermediate blocks introduce specialized 1×1 convolutional reduction layers intercalated with standard 3×3 convolutions. These 1×1 filters reduce the feature space depth (channel dimensions) before expanding it again via deeper layers (e.g., expanding to 512 and 1024 channels), allowing the network to develop highly non-linear feature representations without expanding the parameter count (REDMON et al., 2016).
- **Dense Linear Mapping and Tensor Output Formulation:** The final convolutional layers reduce the spatial layout to a localized 7×7 grid with a depth of 1024

features. This tensor is flattened and passed through a dense linear connection layer (*Conn. Layer*) with 4096 neurons to perform global spatial reasoning. The final output layer constructs a rigid $7 \times 7 \times 30$ tensor.

Mathematical Decoding of the Final Output Tensor

To connect the network architecture to its analytical output, the dimensions of the final $7 \times 7 \times 30$ tensor must be mathematically decoded. The shape is defined by the fundamental formulation:

$$\text{Output Shape} = S \times S \times (B \times 5 + C) \quad (2.2)$$

In the baseline implementation evaluated by Redmon et al. (2016), the image is divided into a grid where $S = 7$. The network is configured to predict $B = 2$ bounding boxes per cell, where each box encapsulates 5 continuous parameters: four spatial coordinates (x, y, w, h) representing the center coordinates, width, and height relative to the cell, and one scalar confidence score. The remaining $C = 20$ elements correspond to the conditional class probabilities for the Pascal VOC dataset. This rigid mathematical output structure proves how a single-shot forward pass successfully maps raw visual attributes straight into complete bounding box coordinates and object classifications concurrently.

2.1.4.4 YOLOv8: Architectural Advancements in Technical Layouts

For this research, YOLOv8 was selected as the core object detection architecture due to its state-of-the-art performance in balancing inference speed and mean Average Precision (mAP). Developed by Ultralytics, YOLOv8 introduces fundamental changes to the traditional YOLO pipeline, transitioning from anchor-based bounding box matching to an anchor-free regression paradigm and implementing an optimized backbone for feature aggregation (JOCHER; CHAURASIA; JING, 2023; TERVEN; CORDOVA-ESPARZA, 2023).

The key architectural advancements that justify its use in complex engineering drawing analysis are detailed below:

- **Anchor-Free Paradigm:** YOLOv8 adopts an anchor-free detection strategy, a major shift from previous versions that relied on predefined anchor box clusters. By predicting the object center and bounding box offsets directly, the model reduces the computational complexity of the prediction head and significantly improves generalization across extreme aspect ratios — a critical factor for detecting highly

elongated dimension lines and small technical symbols (GE et al., 2021; TERVEN; CORDOVA-ESPARZA, 2023).

- **C2f Module and Backbone Efficiency:** The architecture replaces the traditional C3 module with the *Cross-stage Partial Bottleneck with two convolutions* (C2f) module. As illustrated in Figure 3, this component splits and replicates channel paths before concatenation, enhancing feature gradient flow and enriching the representation of spatial hierarchies while maintaining a lightweight computational profile (JOCHER; CHAURASIA; JING, 2023).
- **Decoupled Regression Head:** Following design trends established by YOLOX, YOLOv8 utilizes a decoupled head that processes classification and regression (localization) through completely separate convolutional branches (GE et al., 2021). This approach minimizes the optimization conflict between these two distinct tasks, leading to cleaner gradient descent profiles and faster convergence during the training phase.
- **Advanced Composite Loss Functions:** To guide the learning process when annotations are dense or overlap with geometric primitives, YOLOv8 implements a sophisticated combination of loss functions designed to handle both classification and precise coordinate regression:

- *Box Loss ($\mathcal{L}_{CloU}$):* Bounding box regression is governed by Complete Intersection over Union (CloU) (ZHENG et al., 2020). The CloU metric refines standard IoU calculations by penalizing the normalized distance between box center points and accounting for aspect ratio consistency via the following penalty formulation:

$$\mathcal{L}_{CloU} = 1 - \text{IoU} + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v \quad (2.3)$$

where $\rho(\cdot)$ represents the Euclidean distance between the predicted box center b and ground-truth center b^{gt} , c is the diagonal length of the smallest enclosing bounding box, v quantifies aspect ratio consistency, and α is a positive trade-off parameter (ZHENG et al., 2020). This formulation penalizes spatial discrepancies to ensure the model accurately captures machine views and components, even in cluttered documents.

- *Distribution Focal Loss (DFL):* Operates in tandem with CloU to refine boundaries by optimizing the continuous probability distribution of the box edges (LI et al., 2020). This is particularly relevant in engineering drawings, assisting the model in maintaining localized edge precision when line boundaries are obscured by adjacent text annotations.

- *Classification Loss ($\mathcal{L}_{BCE}$)*: Evaluates the model's ability to assign the correct categorical label to a detected object using Binary Cross Entropy (BCE) across target classes (LIN et al., 2017; JOCHER; CHAURASIA; JING, 2023). Optimizing this metric ensures the model successfully learns to distinguish between technical symbols and orthographic perspectives.

2.1.5 Hyperparameter Tuning

In the context of Deep Learning, model configuration relies on two distinct types of variables: parameters and hyperparameters. **Parameters** (such as weights and biases) are internal variables learned automatically by the algorithm during the training process. In contrast, **Hyperparameters** are external configurations set prior to training that govern the learning process itself and the structure of the model (O'SHEA; NASH, 2015).

Effective hyperparameter tuning is fundamental to the success of neural networks, particularly for complex architectures like Convolutional Neural Networks (CNNs). Common hyperparameters include architectural decisions—such as the depth of the network, kernel size, stride, and zero-padding (O'SHEA; NASH, 2015) — as well as training configurations like batch size, momentum, weight decay, and learning rate schedules.

The selection of these values significantly impacts the model's convergence and generalization capabilities. For instance, an inappropriately high learning rate can cause the model to diverge due to unstable gradients, while a learning rate that is too low may result in slow convergence or stagnation in suboptimal local minima (REDMON et al., 2016). Similarly, regularization hyperparameters, such as dropout rates and data augmentation parameters, are essential for preventing overfitting and ensuring the model generalizes well to unseen data (REDMON et al., 2016). Consequently, the process of optimizing these values — often referred to as hyperparameter tuning — is a critical step in the development pipeline.

2.1.5.1 Key CNN Hyperparameters for Optimization

To ensure optimal model performance, it is necessary to tune the set of critical CNN hyperparameters. A detailed description of the most important ones is provided below:

- **Epochs**: An epoch represents one complete pass of the entire training dataset through the neural network, serving as a primary hyperparameter that dictates the training duration. The selection of the number of epochs is a balancing act: too few iterations may result in **underfitting**, where the model fails to learn the underlying patterns of the data. Conversely, an excessive number of epochs often

leads to **overfitting** — a phenomenon where the model “memorizes” the training data, including its specific noise and outliers, rather than learning generalizable features. This results in high accuracy on training samples but a significant failure to generalize when presented with unseen engineering drawings (BENGIO, 2012).

- **Batch Size:** This parameter defines the number of training samples processed before the model’s internal parameters are updated. Smaller batch sizes require less memory and can offer a regularizing effect due to the noise in gradient estimates, whereas larger batch sizes (e.g., 64) provide more stable gradient estimates but require significant computational resources (BENGIO, 2012).
- **Learning Rate:** The learning rate is arguably the most critical hyperparameter, as it controls the step size at each iteration while moving toward a minimum of a loss function. A learning rate that is too high can cause the training to diverge or oscillate effectively, while a value that is too small can result in a painfully slow convergence process or getting stuck in suboptimal solutions (BENGIO, 2012).
- **Optimizer:** The optimizer is the algorithm or mathematical formula used to minimize the loss function by updating the network’s weights. In this work, three variations were evaluated:
 - *SGD (Stochastic Gradient Descent)*: A classical approach that updates weights using the gradient of the loss function (ROBBINS; MONRO, 1951).
 - *Adam (Adaptive Moment Estimation)*: An algorithm that computes adaptive learning rates for each parameter (KINGMA; BA, 2014).
 - *AdamW*: A variation of Adam that decouples weight decay from the gradient update, often leading to better generalization performance (LOSHCHILOV; HUTTER, 2019).
- **Patience:** This parameter is associated with the “Early Stopping” technique. It defines the number of epochs the training process continues without observing an improvement in a monitored metric (such as validation loss or mAP). If the metric does not improve within the specified “patience” window, training is halted to prevent unnecessary computation and potential overfitting (PRECHELT, 2002).

2.1.5.2 Hyperparameter Optimization with Optuna

Given the high-dimensional search space and the interdependence between the aforementioned hyperparameters, this study utilized **Optuna**, an open-source hyperparameter optimization (HPO) framework (AKIBA et al., 2019). Optuna employs a define-by-run API (Application Programming Interface), allowing for a dynamic construction of the search space.

The optimization process in Optuna is structured around two main concepts:

1. **Study:** A session that aims to optimize a specific objective function (in this case, maximizing the mAP or minimizing validation loss).
2. **Trial:** A single execution of the objective function with a specific set of hyperparameter values.

Optuna utilizes the **Tree-structured Parzen Estimator (TPE)** algorithm as its default sampler. TPE is a Bayesian optimization method that builds a probabilistic model of the objective function, focusing the search on promising regions of the hyperparameter space while maintaining exploration of unknown areas. Additionally, Optuna's efficient pruning feature automatically terminates unpromising trials early in the process, significantly reducing the total computational time required to find the optimal configuration (AKIBA et al., 2019).

Parameter Importance via fANOVA

To quantify the impact of each hyperparameter on the model's performance, Optuna employs the **Functional Analysis of Variance (fANOVA)** framework. Introduced by Hutter, Hoos e Leyton-Brown (2014), fANOVA is a sophisticated method for hyperparameter importance assessment that decomposes the variance of the objective function (e.g., mAP) into components attributable to individual hyperparameters and their interactions.

The algorithm functions by fitting a Random Forest regression model to the data collected during the *Trials*. This surrogate model maps the hyperparameter configurations to their observed performance. fANOVA then marginalizes the predictions of the Random Forest over the hyperparameter space to estimate the "main effect" of each parameter. This statistical approach allows for a robust identification of which parameters are "vital few" and which are "trivial many", providing a data-driven justification for the final model configuration (HUTTER; HOOS; LEYTON-BROWN, 2014).

Interpretability with SHAP Values

While fANOVA identifies which parameters are most influential, it does not inherently describe the direction of their impact. To address this, this study incorporates **SHAP (SHapley Additive exPlanations)**, a game-theory-based approach to explain the output of machine learning models (LUNDBERG; LEE, 2017).

SHAP assigns each hyperparameter an importance value for each trial, representing how much that specific configuration contributed to the deviation from the average performance. By aggregating these values, it is possible to visualize whether higher or lower values of a parameter (e.g., a larger *batch size* or a smaller *learning rate*) are positively or negatively correlated with the model's fitness score. This provides a granular, interpretable view of the hyperparameter landscape, revealing complex synergies between variables that traditional methods might overlook (LUNDBERG; LEE, 2017).

2.1.6 K-Fold Cross-Validation

To ensure the robustness of the results and provide a reliable estimate of the model's generalization performance, this study employed the *K-Fold Cross-Validation* method. This technique is essential for mitigating the bias associated with a single random train-test split, particularly in industrial datasets where specific features or document qualities might disproportionately influence the learning process.

In K-Fold Cross-Validation, the entire dataset is randomly partitioned into K mutually exclusive subsets (or “folds”) of approximately equal size (Figure 5). The evaluation process is then repeated K times. In each iteration, one fold is held out as the validation set, while the remaining $K - 1$ folds are combined to form the training set. The final performance metric, M , is calculated by averaging the results obtained from each of the K iterations:

$$M = \frac{1}{K} \sum_{i=1}^K m_i \quad (2.4)$$

where m_i represents the performance metric (e.g., mAP or F1-Score) obtained in the i^{th} fold. This procedure provides a statistically robust estimate of the model's effectiveness on unseen data (STONE, 1974; KOHAVI, 1995).

The primary objective of implementing this strategy is to assess the **stability and reliability** of the proposed framework. By training and evaluating each model variation multiple times, it is possible to:

- **Detect Overfitting:** Identify if the model is overly sensitive to specific patterns in a small portion of the data, which is critical when dealing with diverse engineering drawing layouts.
- **Reduce Variance:** Ensure that the reported metrics are not the result of a “lucky” data split, but rather a consistent reflection of the architecture's capability.

- **Maximize Data Utility:** In scenarios with limited annotated engineering documents, K-Fold ensures that every data point is used for validation exactly once, providing a more comprehensive evaluation than a traditional hold-out method.

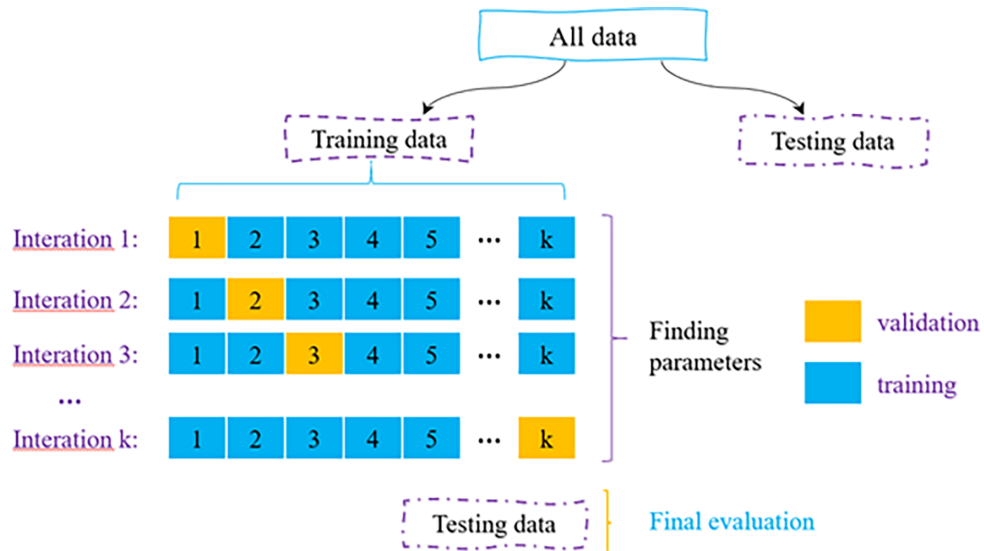


Figure 5 – Visual representation of the K-Fold Cross-Validation process. Adapted from Ngo, Nguyen e Tran (2023).

2.1.7 Optical Character Recognition (OCR)

In many computer vision applications, detecting objects is only the first step for subsequent tasks that often require extracting semantic information such as textual content. Optical Character Recognition (OCR) plays a crucial role in this context. OCR techniques are designed to convert various types of documents — including scanned images, PDF files, and photographs — into editable and searchable data. Historically applied to tasks ranging from digitizing archives to automating administrative workflows, OCR enables the efficient integration of unstructured data into computational systems (MORI; SUEN; YAMAMOTO, 1992). Recent advancements have further expanded these capabilities, fostering innovation across multiple fields (LIN et al., 2023).

In the specific domain of engineering, a technical drawing is more than a collection of graphical shapes, it is a dense repository of critical specifications encoded as text, including dimensions, tolerances, and material notes. While Convolutional Neural Networks (CNNs) excel at identifying geometric structures, they are inherently limited in processing this textual data. Therefore, OCR is employed to bridge this gap. By transforming textual annotations from static pixels into machine-readable strings, OCR complements the geometric understanding provided by object detection, ensuring that the complete set of manufacturing specifications is captured and made available for automated analysis (MORI; SUEN; YAMAMOTO, 1992; LIN et al., 2023).

2.1.8 Natural Language Processing

Natural Language Processing (NLP) represents the intersection of computer science, artificial intelligence, and linguistics. Its primary objective is to enable computational systems to analyze, understand, and derive meaning from human language in both written and spoken forms. This field operates through hierarchical levels of abstraction, progressing from the structural analysis of individual words (morphology) and sentence structure (syntax) to the complex interpretation of meaning (semantics) and contextual usage (pragmatics) (NAGARHALLI; VAZE; RANA, 2021; BROWNLEE, 2017).

In the context of this research, NLP serves as the essential bridge between raw digital data and engineering knowledge. While Optical Character Recognition (OCR) is capable of digitizing visual symbols into text strings, it lacks the capacity for comprehension. NLP techniques are therefore required to parse these strings—for instance, to identify that the character sequence “ $\phi 50$ H7” is not merely alphanumeric data, but a semantic representation of a diameter with a specific tolerance class.

2.1.8.1 Multimodal Large Language Models (MLLMs)

The evolution of Natural Language Processing (NLP) has culminated in the development of Multimodal Large Language Models (MLLMs), which currently represent the state-of-the-art for complex semantic reasoning tasks. Unlike unimodal models, MLLMs are designed to simultaneously process and integrate multiple streams of information, such as text, images, and structured data (DORIS et al., 2024).

In the specific context of technical drawing interpretation, where graphical objects are extracted via CNNs and textual data via OCR, the primary challenge lies in the synthesis of these distinct information streams. A robust system must establish that a specific dimension string (e.g., $\phi 50\text{mm}$) corresponds to a particular geometric primitive detected in the drawing, or that a specific note in the title block dictates the material properties of a component. This necessitates a *Vision-Language* approach, utilizing models capable of reasoning across different modalities to establish precise spatial and semantic correspondence (DORIS et al., 2024).

Modern MLLM architectures, often inspired by frameworks such as Contrastive Language-Image Pre-training (CLIP), learn a shared representation space for both images and text. This shared embedding enables a native understanding of the semantic relationship between visual elements and natural language descriptions. By leveraging the advanced reasoning capabilities of state-of-the-art models like GPT and Gemini, these systems can interpret drawings holistically. This allows for associative reasoning that links localized components to their technical specifications and answers complex queries regarding the engineering design (XUESONG et al., 2025; WANG et al., 2025;

LIANG; WANG; LIU, 2025). Ultimately, this integrated understanding — bridging the gap between raw pixels and technical intent — represents the current frontier of automated drawing interpretation.

2.1.9 Evaluation Metrics

The performance of the proposed pipeline is evaluated using a set of metrics widely recognized in the fields of Computer Vision and Information Extraction. These metrics quantify the model’s ability to precisely localize regions of interest and accurately recognize textual and geometric data.

The following metrics are utilized to assess both extraction and validation stages:

- **Precision:** Measures the proportion of true positives among all instances classified as positive. It indicates how precise the model is in identifying relevant results (POWERS, 2011).

$$\text{Precision} = \frac{TP}{TP + FP}$$

- **Recall:** Measures the proportion of true positives among all instances that actually belong to the positive class. It evaluates the model’s ability to retrieve all relevant cases (POWERS, 2011).

$$\text{Recall} = \frac{TP}{TP + FN}$$

Where TP is True Positives, FP is False Positives, and FN is False Negatives.

- **IoU (Intersection over Union):** A standard metric for object detection tasks. It measures the geometric overlap between the predicted bounding box and the ground truth bounding box. The higher the IoU, the better the localization performance (REZATOFIGHI et al., 2019).

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

- **F1-Score:** The harmonic mean of Precision and Recall. It is particularly useful in situations with imbalanced classes as it provides a single score that balances both false positives and false negatives (POWERS, 2011).

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

- **mAP (Mean Average Precision):** The average of the Average Precision (AP) calculated for all classes. It is the standard metric for evaluating the accuracy of object detectors. In this work, it considers both **mAP@0.5** (IoU threshold of 0.5) and **mAP@0.5:0.95** (average over multiple IoU thresholds), which provides a robust measure of detection performance at different levels of strictness (LIN et al., 2014).

- **Fitness:** A composite metric used (particularly in YOLO frameworks) to optimize hyperparameters during training. It combines detection accuracy and strictness into a single value, typically calculated as a weighted average of mAP metrics (JOCHER et al., 2020):

$$\text{Fitness} = 0.1 \cdot \text{mAP@0.5} + 0.9 \cdot \text{mAP@0.5:0.95}$$

This metric prioritizes models that achieve high precision at stricter IoU thresholds.

2.2 RELATED WORKS

To establish a robust foundation for this research, a systematic literature review (SLR) was conducted to map the current landscape of image-to-text transformation techniques applied to technical engineering drawings. This section details the methodological rigor and the synthesis of the state-of-the-art.

2.2.1 Review Methodology

The systematic literature review was conducted adhering to the protocol proposed by Kitchenham et al. (2007), utilizing the Parsifal online platform (BRERETON; KITCHENHAM; BUDGEN,) to support the planning, selection, and data extraction phases. The study design was structured around the **PICOC strategy** (Population, Intervention, Comparison, Outcomes, and Context), which defined the research scope as follows:

- **Population:** Technical and engineering drawings (2D projects).
- **Intervention:** Image-to-text transformation or image-based reasoning (OCR, Deep Learning, and MLLMs).
- **Comparison:** Manual verification processes or traditional automated approaches.
- **Outcomes:** Precision, processing time, and data extraction efficiency.
- **Context:** Industrial and engineering environments focusing on automated validation.

To guide the investigation, five Research Questions (RQs) were formulated to identify: (i) techniques and tools, (ii) the most effective models, (iii) evaluation metrics, (iv) application domains, and (v) primary challenges reported in the field. The search strategy was executed across five major electronic databases: *ACM Digital Library*, *IEEE Digital Library*, *ScienceDirect*, *Scopus*, and *MDPI* (Multidisciplinary Digital Publishing Institute).

The selection process prioritized primary studies published in English between 2015 and 2025. To ensure the reliability of the results, the selected studies underwent a quality assessment using a checklist of eight criteria, covering aspects such as dataset description, reproducibility, and metric definition. Each study was scored based on its compliance, with a minimum cutoff of 3 points required for inclusion. This rigorous screening process narrowed the initial pool of 125 results down to 12 final articles.

2.2.2 Synthesis of Findings

Based on the established protocol, the twelve selected studies (published between 2020 and 2025) demonstrate a clear technological evolution from traditional Optical Character Recognition (OCR) and Deep Learning-based object detection to emerging multimodal architectures.

The review identified key application domains and standardized evaluation metrics, while highlighting recurring challenges in the field as follows:

- **Data Quality:** Significant inconsistency and visual noise in industrial legacy documents.
- **Structural Complexity:** High density of overlapping geometric and alphanumeric information in 2D drawings.
- **Semantic Limitations:** The inability of conventional models to perform context-aware extraction without visual grounding.

2.2.3 Selected Studies Analysis

The literature on automated information extraction from engineering drawings encompasses a variety of approaches, ranging from traditional computer vision techniques to advanced multimodal deep learning frameworks. To facilitate a comparative understanding of the state-of-the-art, the selected studies were categorized based on their thematic and methodological similarities.

The identification of each study, along with its respective bibliographic reference and title, is provided in Table 2. This mapping serves as the foundation for the thematic discussion presented in this section. Building upon this baseline, Table 3 illustrates the core relationships between the analyzed studies, highlighting how different authors address common challenges such as semantic association and component segmentation.

For instance, it is observed that **Article 1** maintains a direct link with **Articles 8 and 12**, reflecting a common focus on utilizing language models and semantic reasoning

Table 2 – Mapping of the reviewed articles and their corresponding references.

Article	Reference	Title
1	(XUESONG et al., 2025)	Associative reasoning for engineering drawings using an interactive attention mechanism.
2	(MAUPOU et al., 2024)	Automatic raster engineering drawing digitisation for legacy parts towards advanced manufacturing.
3	(GAO; ZHAO; SMIDTS, 2020)	Component detection in piping and instrumentation diagrams of nuclear power plants based on neural networks.
4	(ZHANG et al., 2023)	Component segmentation of engineering drawings using Graph Convolutional Networks.
5	(SCHÖNFELDER et al., 2024)	Deep learning-based text detection and recognition on architectural floor plans.
6	(HU; ZHANG; LIANG, 2021)	Detection of surface roughness of mechanical drawings with deep learning.
7	(SCHEIBEL; MANGLER; RINDERLE-MA, 2021)	Extraction of dimension requirements from engineering drawings for supporting quality control in production processes.
8	(KHALLOULI et al., 2024)	Leveraging Transformer-Based OCR Model with Generative Data Augmentation for Engineering Document Recognition.
9	(TORO; TARKIAN, 2025)	Optimizing Text Recognition in Mechanical Drawings: A Comprehensive Approach.
10	(SCHLAGENHAUF; NETZER; HILLINGER, 2023)	Text detection on technical drawings for the digitization of brown-field processes.
11	(XU et al., 2024)	Tolerance Information Extraction for Mechanical Engineering Drawings—A Digital Image Processing and Deep Learning-based Model.
12	(LIANG; WANG; LIU, 2025)	Towards a self-cognitive complex product design system: A fine-grained multi-modal feature recognition and semantic understanding approach using large language models in mechanical engineering.

to interpret technical documents. Conversely, **Article 2** connects to **Articles 10 and 11**, which are primarily dedicated to the digitization and extraction of technical tolerances. Such groupings reinforce a clear trend of specialization within the field, where studies

Table 3 – Cross-analysis of the selected studies and their thematic relationships.

Thematic Relationships	Study ID
Integration of Multimodal Large Language Models (MLLMs) and multimodal semantic association for context-aware extraction.	Articles 1, 8 and 12
Focused on the digitization of legacy raster drawings, specifically targeting technical tolerances and text annotations.	Articles 2, 10 and 11
Application of CNNs for the localized detection of mechanical and technical components.	Articles 3, 4 and 6
Advanced segmentation techniques for decomposing rasterized engineering projects into functional layers.	Articles 2, 4 and 11
Development of specialized OCR pipelines for text detection in blueprints and architectural documents.	Articles 5, 8 and 9
Automated workflows for the systematic extraction of dimensions and geometric tolerances from complex projects.	Articles 7, 9 and 11

coalesce into well-defined technical subsets within the engineering drawing domain.

2.2.3.1 Thematic Discussion and Evolution

Recent reviews, such as the one conducted by Jang, Roh e Lee (2025), reveal a rapid adoption of generative artificial intelligence (Gen AI) from 2021 onwards. Their analysis of 161 studies shows a transition from Generative Adversarial Networks (GANs) to transformer and diffusion-based models. In the domain of product design, Wang et al. (WANG et al., 2025) proposed the Cognitive Intelligence-enabled Product Design (CIPD) paradigm, which integrates semantic understanding and multimodal generation through LLMs and cognitive computing, highlighting the relevance of human-AI collaboration in technical drawings.

Historically, earlier reviews by Moreno-García, Elyan e Jayne (2019) emphasized the role of CNNs in digitization but pointed out the persistent scarcity of annotated datasets. Following this evolution, specific aspects have been addressed: Maupou et al. (2024) focused on tolerance extraction in legacy parts; Gao, Zhao e Smidts (2020) and Zhang et al. (2023) explored component detection and segmentation in P&IDs and rasterized drawings using neural networks and Graph Convolutional Networks (GCNs).

In the context of quality control and BIM (Building Information Modeling), Scheibel, Mangler e Rinderle-Ma (2021) and Schönfelder et al. (2024) developed methods

for dimension extraction and text recognition in floor plans. More recently, the focus has shifted toward transformer-based OCR with generative data augmentation (KHALLOULI et al., 2024) and multimodal semantic understanding using LLMs to interpret complex mechanical drawings (LIANG; WANG; LIU, 2025). Optimized pipelines for mechanical drawings, such as those by Toro e Tarkian (2025) and Xu et al. (2024), combine digital image processing with deep learning to extract tolerance information more effectively.

The reviewed literature highlights steady advances in applying AI to engineering drawing analysis. Nonetheless, persistent challenges remain, including the limited availability of annotated datasets and the complexity of integrating textual and geometric information in a unified analytical framework—a challenge that this research addresses through a multi-stage pipeline.

2.2.4 Reported Challenges

The literature review reveals a series of technical, methodological, and practical challenges encountered in the extraction of textual and semantic information from technical engineering drawings. These challenges are particularly acute in industrial contexts, where the diversity of formats, the presence of legacy documents, and visual complexity hinder process automation. The primary issues observed in the analyzed articles are presented below, organized by theme.

One of the most recurrent challenges relates to **data quality and availability**. The scarcity of public, annotated datasets specific to technical drawings was highlighted in several studies (MAUPOU et al., 2024; GAO; ZHAO; SMIDTS, 2020; SCHÖNFELDER et al., 2024; HU; ZHANG; LIANG, 2021; SCHEIBEL; MANGLER; RINDERLE-MA, 2021; TORO; TARKIAN, 2025). The absence of such datasets compromises the training of robust Deep Learning models, especially when seeking generalization across varied industrial scenarios. Furthermore, documents digitized from paper sources, often in low resolution or containing visual noise, further complicate the application of OCR models (SCHÖNFELDER et al., 2024; HU; ZHANG; LIANG, 2021; SCHLAGENHAUF; NETZER; HILLINGER, 2023).

Text detection and recognition also represent a significant barrier. Text in drawings appears in various orientations (horizontal, vertical, inclined), reduced sizes, superimposed on lines and symbols, or even handwritten, factors that limit the performance of conventional OCR models (XUESONG et al., 2025; GAO; ZHAO; SMIDTS, 2020; SCHÖNFELDER et al., 2024; HU; ZHANG; LIANG, 2021; KHALLOULI et al., 2024; TORO; TARKIAN, 2025). In many cases, models fail to correctly identify dimensions, special characters, or degraded text. Additionally, frequently fragmented text (such as dimensions and tolerances) requires semantic regrouping processes (SCHEIBEL; MANGLER; RINDERLE-MA, 2021).

Another critical point is the **detection of small objects and technical symbology**, which affects the performance of computer vision algorithms, particularly when employing traditional segmentation or template matching techniques (MAUPOU et al., 2024; GAO; ZHAO; SMIDTS, 2020; ZHANG et al., 2023; HU; ZHANG; LIANG, 2021). Components such as arrows, roughness symbols, or dimension lines may be ignored or confused with other graphical elements. When these elements are isolated or appear in unusual orientations, detection becomes even more challenging.

Difficulties related to **representation and semantic extraction** also stand out. Many models prioritize visual characteristics, such as geometry and topology, neglecting the semantic relationship between text and graphical elements (XUESONG et al., 2025; MAUPOU et al., 2024; SCHEIBEL; MANGLER; RINDERLE-MA, 2021; KHALLOULI et al., 2024). This compromises the global understanding of the drawing, especially in tasks demanding associative reasoning or contextualized information retrieval.

Another frequent challenge is **grouping and spatial mapping**, which is fundamental for correctly relating textual annotations to their corresponding geometric entities (XUESONG et al., 2025; GAO; ZHAO; SMIDTS, 2020; SCHEIBEL; MANGLER; RINDERLE-MA, 2021). Variations in orientation, the distance between elements, and the presence of irrelevant text in close proximity hinder the creation of reliable links, particularly in drawings with low standardization.

From a computational perspective, the **high processing cost** involved in applying advanced models, such as Transformers, is notable, especially when approaches like sliding windows are used to handle large images (KHALLOULI et al., 2024; SCHLAGENHAUF; NETZER; HILLINGER, 2023). Furthermore, multimodal or Natural Language Processing (NLP) based models face domain limitations, demonstrating difficulties in correctly interpreting specific engineering technical information (LIANG; WANG; LIU, 2025).

Finally, issues such as **data privacy** and **dependence on external servers** (in solutions based on Large Language Models) raise concerns regarding the practical application of these technologies in sensitive industrial environments (TORO; TARKIAN, 2025).

In summary, the analyzed studies indicate that while advances in Deep Learning and multimodal models have expanded the possibilities for automated information extraction in technical drawings, significant gaps remain regarding data availability, document visual complexity, model generalization, and the integrated semantic interpretation of text and image.

2.2.5 Positioning and Contributions of this Work

The literature review and the analysis of reported challenges highlight a clear gap in current methodologies: while traditional OCR and Computer Vision techniques have advanced, they often struggle to integrate visual detection with semantic understanding in complex industrial scenarios. Specifically, the challenges of **detecting small technical symbology**, **handling unit inconsistencies**, and **interpreting the semantic context** of dimensions remain significant hurdles (MAUPOU et al., 2024; LIANG; WANG; LIU, 2025).

This work addresses these gaps by proposing a novel, multi-stage pipeline that integrates state-of-the-art in **Object Detection** with **CNNs (YOLO)** and **Semantic Interpretation** with **MLLMs**. Therefore, unlike approaches that rely solely on global OCR or traditional segmentation, which often fail due to the visual clutter and complex layout of engineering drawings, this work utilizes a dedicated object detection layer to precisely localize regions of interest (such as dimension lines and tolerance blocks) prior to character recognition.

Besides that, it tackles the semantic interpretation gap identified in recent studies (KHALLOULI et al., 2024; LIANG; WANG; LIU, 2025) by employing Prompt Engineering techniques to normalize and validate extracted data. This allows the solution to not only read text but to reason about it, resolving unit conflicts (e.g., inches vs. millimeters) and filtering false positives caused by data artifacts. By validating this approach on a real-world dataset from a large-scale manufacturing environment, this work provides empirical evidence of how hybrid AI architectures can overcome the limitations of legacy document digitization.

3 PROPOSED APPROACH

This chapter describes the reasoning used to solve the challenges addressed in the process of extracting, understanding, and evaluating the machine dimensional data presented in engineering drawings.

3.1 CONCEPTUAL FRAMEWORK

The proposed framework integrates localization, semantic reasoning, and digitization to transform engineering drawings into structured and semantically contextualized information, enabling the automated verification of dimensional correctness.

Conceptually, the framework is organized into three complementary layers. The localization layer identifies the relevant engineering entities within the drawing through a hierarchical object detection strategy. The semantic reasoning layer interprets the engineering meaning of the localized regions by exploiting their spatial context and orthographic perspective. Finally, the digitization layer extracts the numerical information from the previously localized dimensions using Optical Character Recognition (OCR), enabling comparison with the corresponding semantic labels and the reference values stored in the Industrial Production System database.

To implement these conceptual layers, the proposed architecture is organized into a four-stage processing pipeline, illustrated in Figure 6. The localization layer is implemented by four object detection models distributed across two sequential stages. Two models operate in Stage 1 to detect machine views and classify their orientations, while two additional models refine the localization by detecting individual dimensions and tolerances in Stage 2. Stage 3 implements the semantic reasoning layer through a Multimodal Large Language Model (MLLM), which interprets the engineering meaning of the localized views. Finally, Stage 4 performs digitization by applying OCR to the localized dimension regions and combines the extracted numerical values with the semantic interpretation to execute the final verification process.

The operational sequence is detailed bellow:

Stage 1 - Global Contextualization: The process initiates with the ingestion of the technical Document, i.e., 2D engineering drawing. Two sequential Object Detection (OD) blocks are applied to establish the spatial foundation. The first block performs **View Detection** (Model 1), identifying the primary machine projections, while the second block handles **Orientation Classification** (Model 2), defining the orthographic perspective (Front, Side, or Top) required for contextual reasoning.

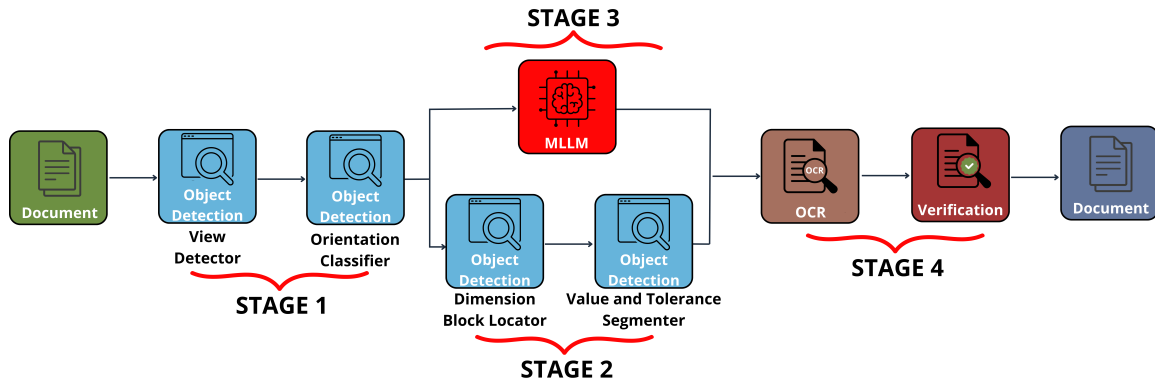


Figure 6 – Schematic of the proposed multi-stage automated workflow.

Stage 2 - Fine-Grained Localization: Following the global localization, the pipeline branches into two tasks in parallel. In this stage, two additional OD blocks are utilized to perform a “zoom-in” operation on the technical features. This involves the precise **Localization of Dimensions and Tolerances** (Models 3 and 4), isolating these specific graphic primitives from the rest of the drawing’s geometry to minimize noise for the subsequent OCR phase.

Stage 3 - Multimodal Semantic Reasoning: Operating in a parallel branch to Stage 2, the **Multimodal Large Language Model (MLLM)** acts as the “cognitive core” of the system. It receives the localized views and, through prompt engineering, identifies which engineering variables (e.g., “Shaft Diameter” or “Hole Depth”) are represented based on their position and context, rather than just reading numerical strings.

Stage 4 - Extraction and Dual-Verification: The outputs from the previous stages converge in this final phase. First, an **OCR (Optical Character Recognition)** engine is applied to the high-resolution crops generated in Stage 2 to extract lexical numerical values. Subsequently, a **Verification** component performs a dual-verification logic, cross-referencing the OCR’s lexical output with the MLLM’s semantic intent. Only data that passes this comparative audit is validated for the final output, effectively neutralizing MLLM hallucinations.

3.2 PROPOSED WORKFLOW IN DETAILS

The complete proposed workflow integrates four trained models with a Multimodal Large Language Model (MLLM) to perform an end-to-end analysis of a technical drawing and validation of dimensional information. Therefore, the automated workflow proceeds as follows:

1. **Data Ingestion:** A machine drawing document is downloaded from the Production System (Figure 7), as well as its existing dimension data. According to the industry

policy, the dimensional data provided by the Production System is intended to be up-to-date and thus will be considered as the “ground truth” for comparison with the engineering drawing.

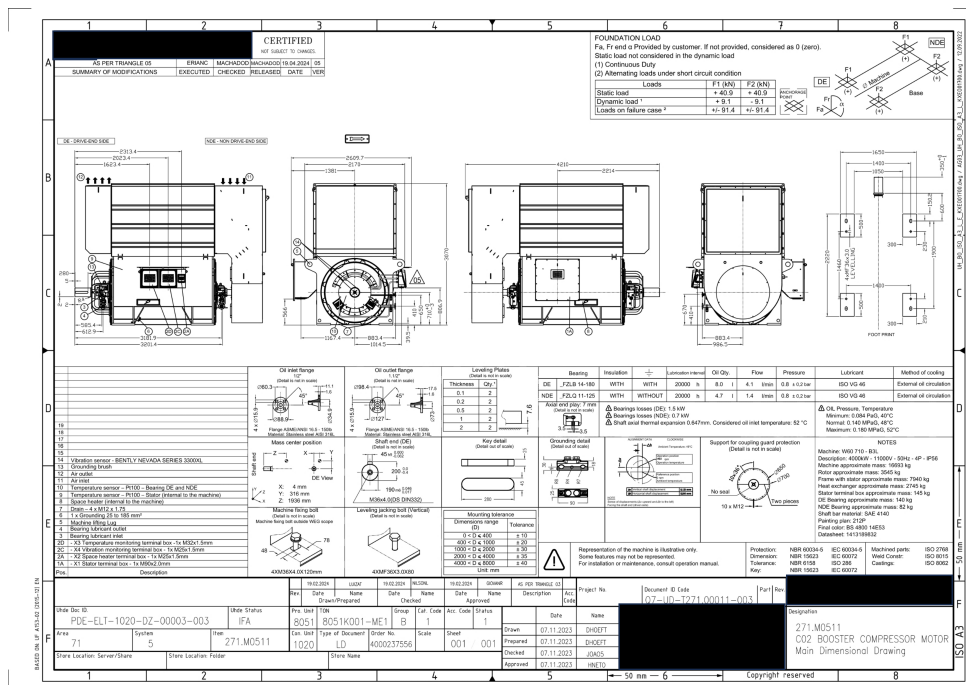


Figure 7 – Example of the source engineering document ingested by the system.

2. **Object Detection Model 1 - View Detection:** It is applied to detect the bounding boxes of all distinct machine views present in the drawing (Figure 8).

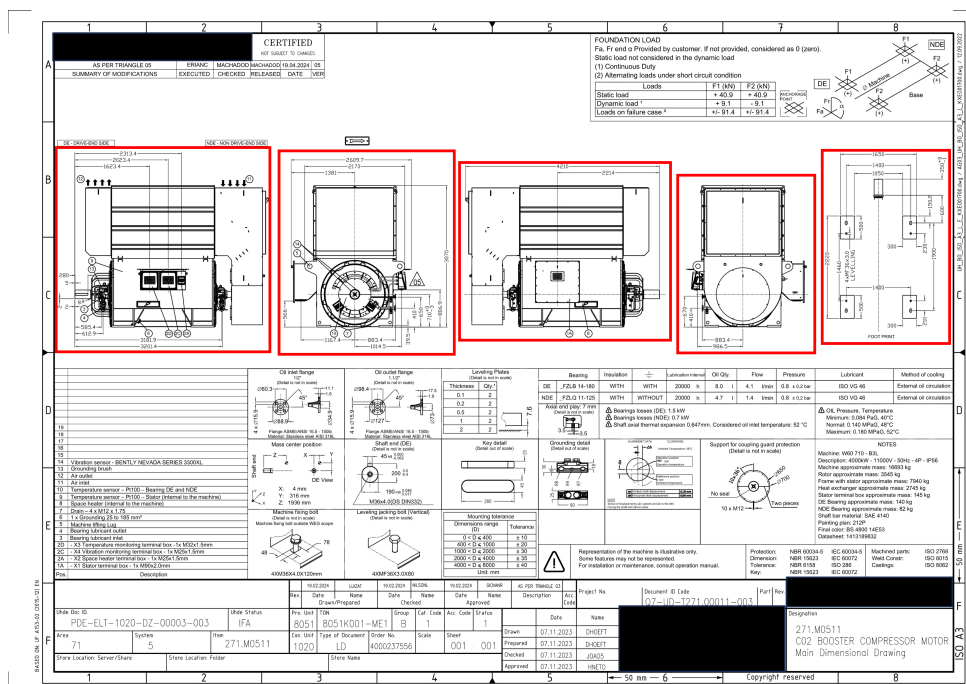


Figure 8 – Application of Model 1 to identify all machine view regions.

3. **Object Detection Model 2 - View Type Classification:** It classifies the type of each detected view (e.g., “Front View”, “Section A-A”, “Top View”, and so forth) (Figure 9).

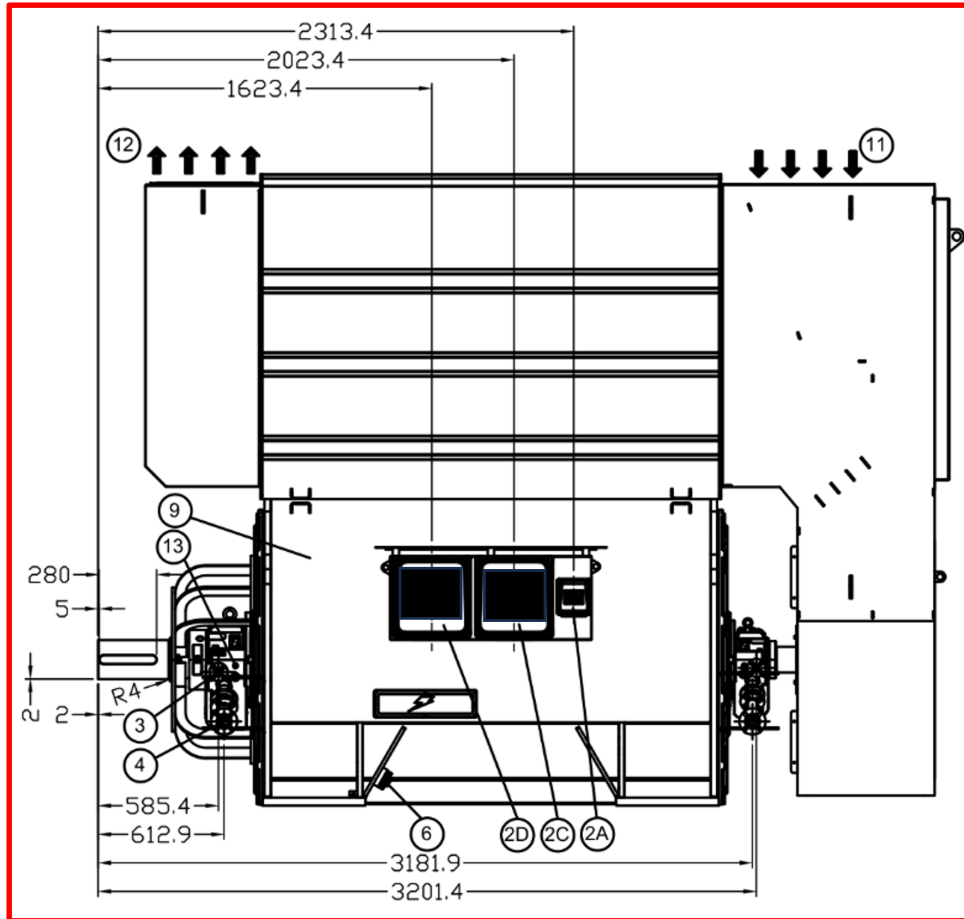


Figure 9 – Application of Model 2 to classify the detected view as “Side View”.

4. **Image and Prompt Generation for MLLM:** Then, with the detected machine views, a new composite image is generated. This image contains the cropped machine views, each annotated with its classified type (Figure 10). This visual information, along with a structured prompt explaining the semantic context of each view type, with the expected dimension to be found in each view, is sent to the MLLM to perform the semantic recognition of the dimensions presented in the views (Appendix A.1 A.2).

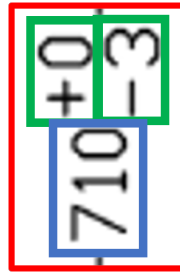


Figure 12 – Application of Model 4 to separate the dimension (blue) from the tolerance (green).

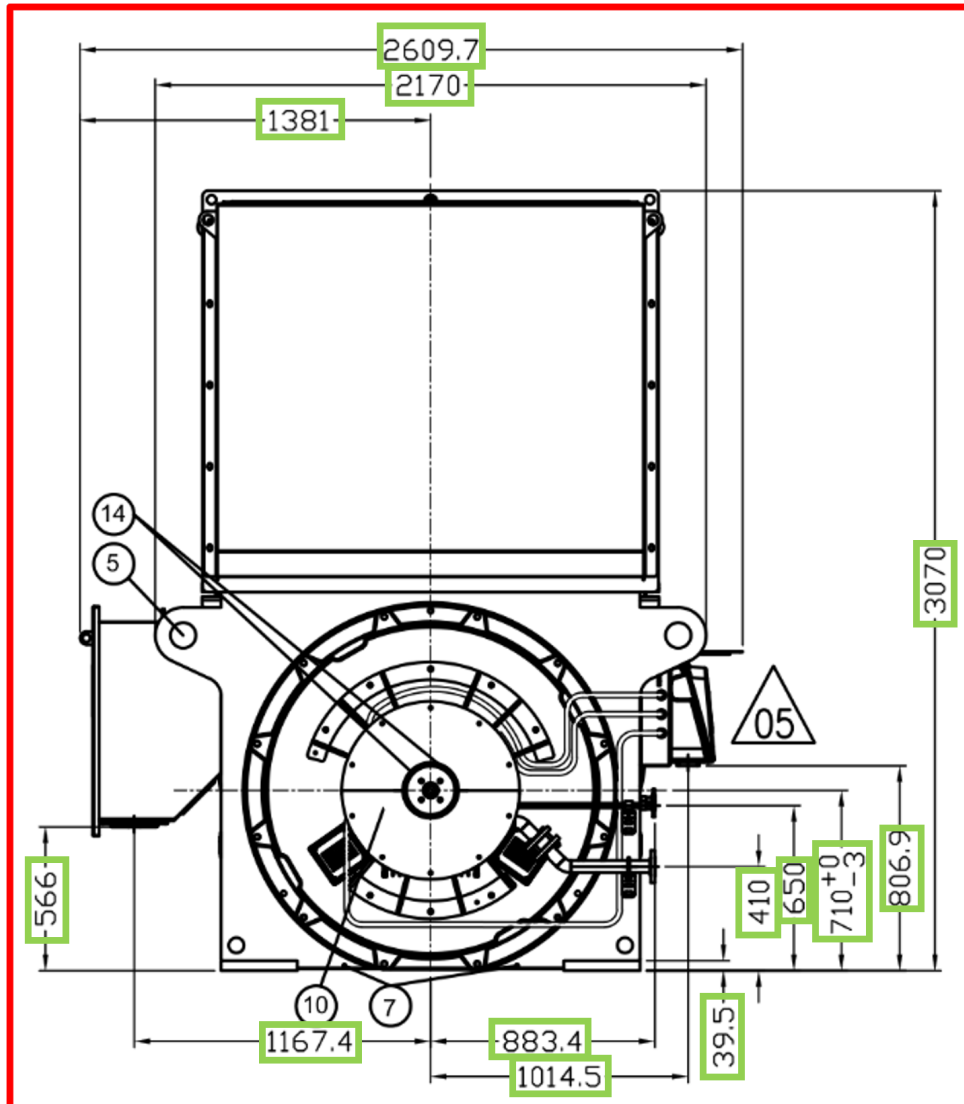


Figure 11 – Application of Model 3 to detect all dimension bounding boxes.

8. **OCR Extraction and Contextual Search:** A standard Optical Character Recognition (OCR) engine extracts the raw text from the high-resolution bounding boxes isolated by Model 4. Rather than performing a blind text extraction, the system executes a contextualized search algorithm. If a discrepancy is flagged during the MLLM data reconciliation phase—recorded in the $\{[system_value, MLLM_value, view_number]\}$ structure—the OCR engine uses the specific `view_number` as a

spatial filter. For instance, if the system expects a nominal value of 50 but the MLLM proposes 49, the algorithm confines its text verification strictly to the bounding boxes belonging to that designated machine view. This localized verification prevents cross-view data contamination, significantly reducing false positives by ensuring that numerical updates are mathematically and spatially coherent with the target engineering projection.

9. **Final Result Synthesis and Visualization:** The pipeline culminates in a comparative analysis block that cross-references the semantic predictions from the MLLM with the lexical text extracted via OCR. In the event of a structural or numerical divergence between these two processing streams, the algorithm dynamically captures the variance and renders a diagnostic metadata panel in the upper-left corner of the final document canvas (Figure 13). This real-time notification ensures that the engineering auditor is immediately alerted to specific dimensional discrepancies before examining the drawing body.

As illustrated in Figure 13, each entry populated within this diagnostic panel follows a structured sequence:

- a) **Measure Name:** The standardized identifier of the target dimension (e.g., AD, AL, BB, BD, or L).
- b) **Database Information (A):** The ground-truth reference value retrieved from the company's Production System database.
- c) **MLLM Identification (G):** The numerical value identified by the semantic and cognitive reasoning of the vision-language model.
- d) **Specific Machine View (V):** The localized orthographic view (Front, Side, or Top) where the engineering primitive was detected.

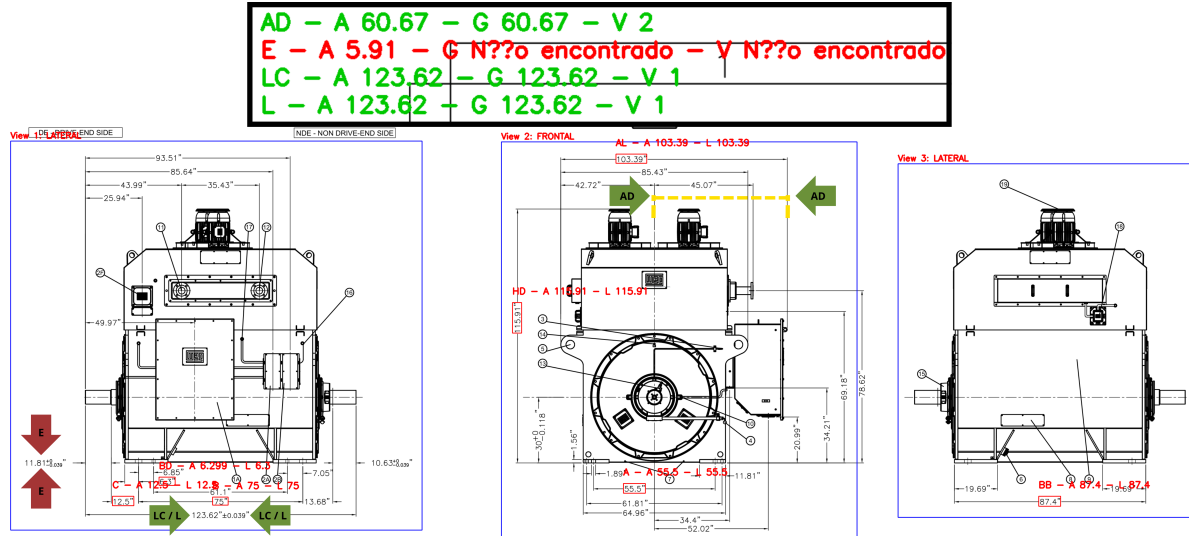


Figure 13 – Verification overlay rendered in the top-left corner of the engineering drawing. Green entries indicate dimensions successfully validated against the reference database, either through direct detection or by semantic and mathematical inference when the dimension is not explicitly present in the drawing (e.g., $AD = AL - AD1$). Red entries indicate dimensions that could not be automatically validated because the system either did not detect the corresponding annotation despite it being present in the drawing (e.g., E), or because the value extracted from the drawing differs from the reference database. Additionally, some dimensions exhibit semantic relationships, such as L and LC , where LC generally represents the overall length while L corresponds to a related dimension; depending on the drawing, both values may be identical or only one of them may be explicitly provided.

The synthesized results from the diagnostic panel are mapped back onto the technical document, generating a graphical overlay for user validation (Figura 14). To facilitate immediate interpretation by the engineering auditor, the framework implements a strict binary color-coding logic across both the metadata panel and the drawing's localized bounding boxes:

- **Green Highlight (Match):** Indicates a perfect validation consensus. This state is reached only when the reference database value (A), the semantic identification by the MLLM (G), and the raw OCR text extraction align seamlessly without discrepancies.
- **Red Highlight (Discrepancy / Not Found):** Indicates a processing conflict or informational gap requiring manual human-in-the-loop review. This state is triggered either by a character mismatch between the extracted text and the reference database, or when the pipeline completely fails to locate the dimension within the views' projections.

In scenarios where a specific engineering measure cannot be located or resolved by the models, the fields for the MLLM value (G) and its respective View (V) are structurally flagged as “Not found” in **RED**. This explicit marking alerts the operator that the parameter could neither be cross-referenced nor visualized within the 2D drawing.

4 DEVELOPMENT METHODOLOGY

This chapter outlines the methodology developed for constructing and evaluating the proposed workflow. As engineering drawings are complex by nature, often containing a wide range of annotations, components, and details, as seen in Figure 7, the methodology is organized into three main sections:

- **Database Preparation:** Addresses the process for data collection, annotation, and creation of balanced subsets for training the models.
- **Model Training:** Describes the training procedures for the four specialized models that compose the detection pipeline:
 1. **View Detector (Model 1):** Localizes and counts the primary machine views within the full drawing layout.
 2. **Orientation Classifier (Model 2):** Categorizes the detected views into Front, Side, or Top perspectives to provide spatial context.
 3. **Dimension Block Locator (Model 3):** Isolates the regions containing dimensional annotations within each individual machine view.
 4. **Value and Tolerance Segmenter (Model 4):** Performs fine-grained extraction, separating nominal values from their respective tolerances.
- **Evaluation:** The process of selecting the most suitable metrics for assessing the performance of each model and the overall workflow.

The proposed pipeline was implemented using the Python programming language, version 3.10.11. The architecture integrates specialized computer vision models, an OCR engine, and a multimodal reasoning layer to ensure robust data extraction. The core components of the system rely on the following technologies:

- **Object Detection Framework:** YOLOv8 was utilized as the base architecture for all detection steps (Models 1 through 4), providing the high-speed inference required for real-time document processing.
- **OCR Engine:** EasyOCR version 1.7.1 was utilized for the extraction of alphanumeric strings from the localized dimension blocks identified in the previous stages.

- **Reasoning Model:** The **Gemini 2.5 Pro** MLLM provides the semantic layer required to interpret engineering intent and validate extracted data against system databases.

The computational tasks, including the hyperparameter optimization, the K -fold cross-validation of the CNN models, the multimodal inference cycles, and the experimental phase, were executed on a dedicated workstation. The training pipeline was processed entirely via Central Processing Unit (CPU), utilizing the following hardware and software environment:

- **Processor:** Intel® Core™ i7-10700 (2.90 GHz, 16 Threads).
- **System Memory:** 32 GB of RAM.
- **Deep Learning Backend:** PyTorch version 2.2.1 (CPU compute mode).
- **Detection Framework:** Ultralytics YOLOv8 (version 8.0.210).

4.1 DATABASE PREPARATION

The foundation of this work is a proprietary database of technical engineering drawings. This section details the systematic process of selection, large-scale labeling, and the quality assurance protocols established for model training.

4.1.1 Dataset Selection and Temporal Scope

The dataset comprises 4,506 technical drawings, spanning from February 22, 2008, to May 21, 2024. This timeframe was strategically selected to capture a diverse range of machine designs, legacy drafting standards, and various document revisions, ensuring the model’s robustness against the “unstructured” nature of legacy industrial data discussed in Chapter 1.

4.1.2 Collaborative Labeling and Gamification

In the development of Artificial Intelligence systems, data-related tasks—such as collection, cleaning, and annotation—typically account for over 80% of the total project effort (SAMBASIVAN et al., 2021; PRESS, 2016). Therefore, to address this substantial bottleneck and ensure the scalability of the annotation process, a proprietary, custom-built **Gamification Tool** was developed in this work (Figure 15).

The primary objective of this tool was to democratize the intensive labeling workload while maintaining volunteer engagement through gamified feedback loops. A total of **4,416 documents** were labeled through a collaborative effort involving **over 200**

participants distributed across strategic global hubs, including Brazil, Mexico, China, and India.

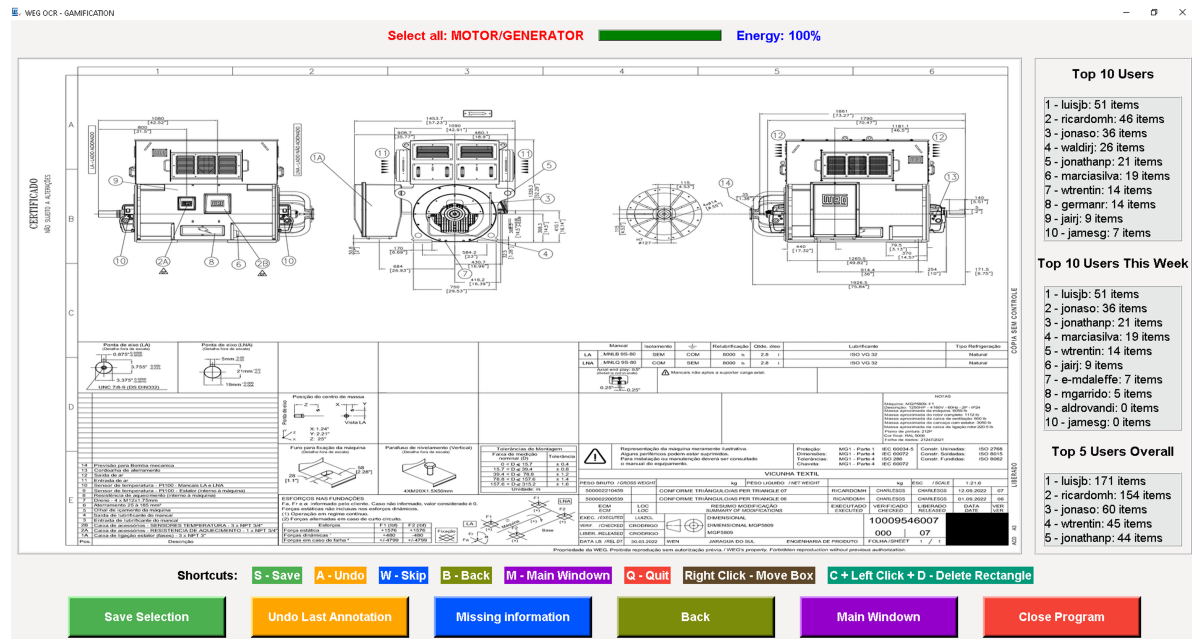


Figure 15 – Interface of the custom-built Gamification Tool used for large-scale annotation of machine views, highlighting the document rendering viewport, real-time competitive leaderboards, and interactive control panels.

4.1.2.1 User Onboarding and Operational Modes

To ensure data consistency, all participants underwent standardized training on the tool's interface and the technical criteria for identifying electric machine views. The tool offered two distinct modes of interaction to accommodate different user preferences:

- **Competitive Mode:** A time-pressured environment designed for high-throughput labeling.
- **Analytical Mode:** A self-paced environment focused on meticulous inspection of complex layouts.

4.1.2.2 Core Labeling Loop and Ergonomic Controls

During the active labeling cycle, the interface dynamically displays a target class prompt at the top of the viewport (e.g., "*Select all: MOTOR/GENERATOR*"). The user performs a click-and-drag action to instantiate a two-dimensional bounding box over the spatial boundaries of the target entity.

To ensure maximum operational efficiency and minimize repetitive strain during long sessions, the software implements a comprehensive suite of mechanical buttons and physical keyboard shortcuts:

- **Annotation Commitment (S / Green Button):** Confirms and commits the active bounding box selection to the local database, automatically updating the user's score.
- **Correction and Rollback (A / Orange Button):** Triggers an immediate rollback of the last annotation, allowing users to quickly redraft misaligned bounding boxes.
- **Bypassing and Document Skip (W / Blue Button):** Used when the prompted target entity is entirely absent from the rendered drawing, registering a formal null-positive state in the dataset.
- **Advanced Spatial Editing:** Users can adjust box locations dynamically via **Right-Click and Drag**, or perform surgical deletions of specific rectangles using the compound command **C + Left-Click + D**.

4.1.2.3 Competitive Mechanics and Quality Assurance

As depicted in the right-hand panel of Figure 15, the tool incorporates social-proof and gamification mechanisms to sustain user engagement over extended periods. The system continuously queries the database to generate three independent leaderboards, each highlighting a different performance metric:

- *Top 10 Users* – ranks users according to the highest number of annotations completed in a single session;
- *Top 10 Users This Week* – ranks users based on the highest number of annotations completed in a single session during the current week;
- *Top 5 Users Overall* – ranks users according to the cumulative number of annotations completed across all sessions.

4.1.2.4 Quality Assurance Labeling

To guarantee the high precision required for industrial-grade computer vision models, the dataset annotation workflow was executed exclusively by engineering professionals with domain expertise in electrical machinery and technical drafting. Rather than relying on crowd-sourced laypersons, all participants were recruited from engineering departments across global operations, ensuring an intrinsic cognitive baseline regarding motor architectures, orthographic projections, and CAD standards.

To enforce inter-annotator consistency and eliminate subjective interpretation across different strategic hubs, a mandatory, standardized training protocol was conducted prior to active labeling sessions. During this onboarding phase, participants

underwent structured instruction on the precise spatial delimitation criteria for target classes.

4.1.3 Hierarchical Labeling Workflow

The dataset annotation process was structured into a four-stage hierarchical workflow, transitioning progressively from macro-level layout classification down to fine-grained technical feature extraction. While the initial stage—focused on global view detection (Model 1)—leveraged the gamification platform to handle high-volume processing across a broad volunteer base, the subsequent stages demanded a significantly higher degree of domain precision, analytical rigor, and meticulous spatial alignment.

To address the elevated complexity of these micro-level tasks, the annotations for Models 2, 3, and 4 (comprising projection orientation classification, dimension block localization, and granular value/tolerance segmentation, respectively) were restricted exclusively to a dedicated core team of **three specialized engineering**.

Due to the extreme visual density, overlapping CAD geometry, and tight bounding-box tolerances inherent in electrical machine drawings, relying on a crowd-sourced approach for these advanced stages would introduce unacceptable inter-annotator variance. By entrusting the fine-grained segmentation to a small, highly qualified triad of domain specialists, the annotation protocol ensured absolute consistency across the dataset, eliminated ambiguity in boundary definitions, and established a rigorous ground truth benchmark essential for high-precision computer vision models.

4.1.3.1 Step I: Electric Machine View Counting

Following the gamification effort, the dataset was categorized into distinct classes representing the layout complexity of each document. This classification is defined by the total number of distinct machine views (N) present in a single drawing:

- **Class 3:** Documents containing exactly three distinct views.
- **Class 4:** Documents containing exactly four distinct views.
- **Class N :** Documents containing N distinct views, where $N \geq 5$.

An analysis of the class distribution, represented by the Pareto chart in Figure 16, reveals a significant imbalance. Classes 3 and 4 represent the majority of the dataset, reflecting standard industrial drafting practices. This distribution was instrumental in guiding the data augmentation strategies and the construction of balanced training subsets.

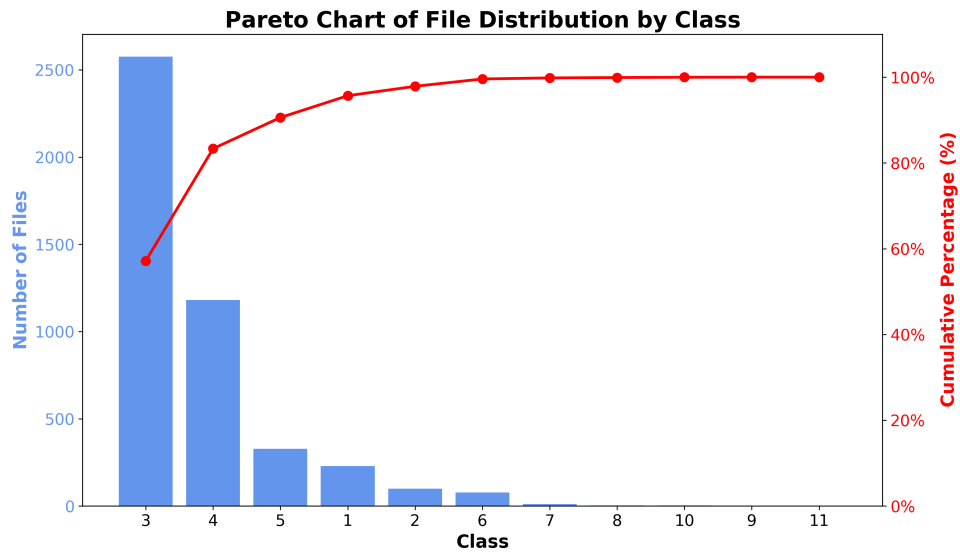


Figure 16 – Pareto distribution of images across different electric machine view classes.

4.1.3.2 Step II: Spatial Orientation Labeling

In the second step, a subset of **1,455 isolated view images** was selected for granular annotation. This specific volume does not represent the total number of engineering source documents, but rather the cumulative sum of individual orthographic views cropped by Model 1 from drawings classified strictly within Class 3 and Class 4. Since these two classes reflect standard industrial drafting practices and contain precisely three or four views per document layout, their sub-components were isolated to build a robust, balanced dataset for orientation training.

A team of three engineering specialists performed the labeling to identify the spatial orientation of each isolated machine view. Each cropped image was classified into one of three primary perspectives: **Front, Side, or Top** (Figure 17). This step is critical for providing the Multimodal Large Language Model (MLLM) with the necessary spatial context to interpret dimensions correctly, ensuring that the model understands the structural alignment of the component before executing semantic data extraction.

4.1.3.3 Step III: Dimension Block Localization

The third step shifted the annotation focus from global perspective analysis to localized technical features, targeting the identification of dimension blocks. For this task, a subset of **2,500 isolated view images**—originally detected and cropped during the Step I pipeline—was audited and granularly labeled by the specialist team (Figure 18).

The primary objective of this step was to generate precise bounding boxes around all alphanumeric dimensional data, effectively decoupling these annotations from the structural and geometric lines of the machine components. Isolating these

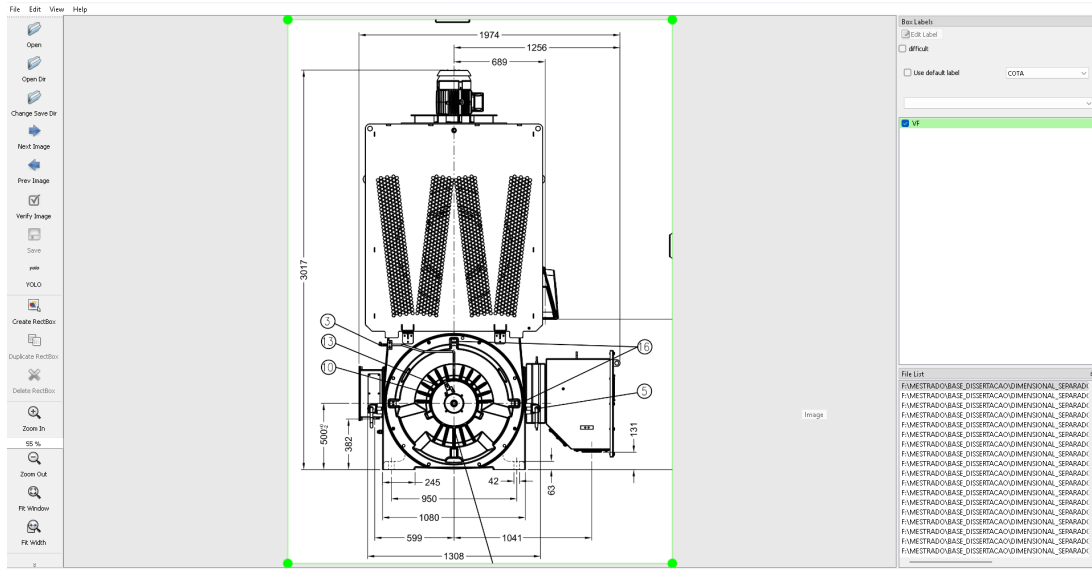


Figure 17 – Labeling Step II: Identification of Front, Side, and Top views to establish spatial context.

textual primitives is essential to reduce visual noise and entropy within the image tensors, establishing an optimized dataset for the subsequent semantic extraction and character recognition phases.

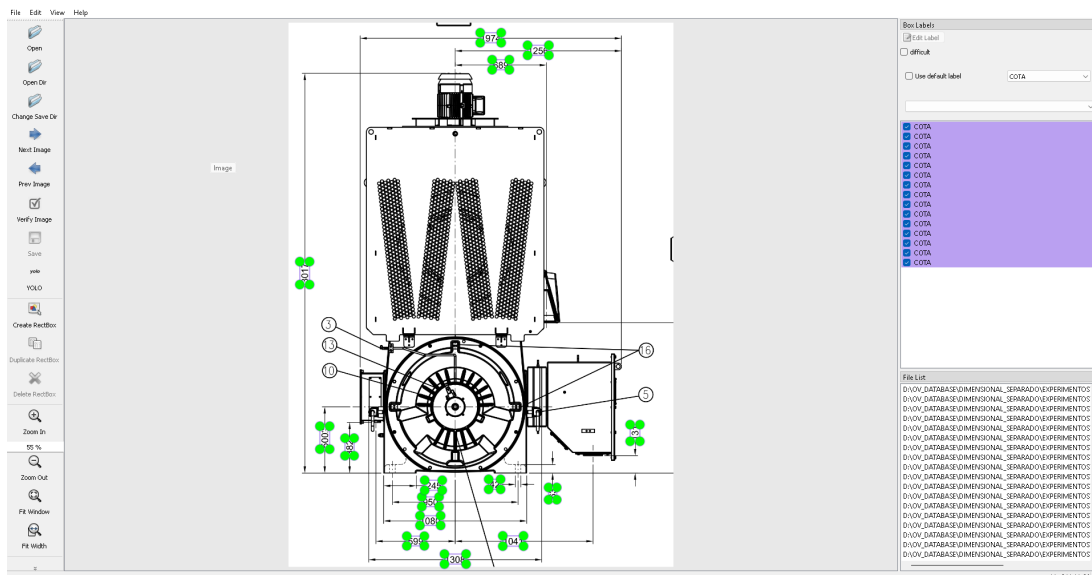


Figure 18 – Labeling Step III: Bounding box localization of dimension blocks within isolated machine views.

4.1.3.4 Step IV: Fine-Grained Dimension and Tolerance Decomposition

The final and most granular annotation step focused on parsing the internal lexical and structural anatomy of the isolated dimension blocks. For this task, a dataset consisting of **3,938 dimension block crops**—directly extracted from the bounding box outputs validated in Step III—was compiled and labeled by the engineering specialists (Figure 19).

To ensure high generalization and robust boundary convergence during the training of the segmentation head, this subset was balanced with a strict operational criterion: half of the selected crops contained explicit technical tolerances, while the remaining half consisted strictly of nominal dimensions without variation limits. This structural variation was paramount to train the model to correctly identify the absence of tolerances, mitigating false-positive segmentations in dense layouts.

Specialists decomposed each dimensional entity into two distinct sub-bounding boxes:

- **Nominal Value:** The primary numerical characters representing the baseline geometric dimension.
- **Tolerance:** The specific alphanumeric string defining the permissible limit of mechanical variation (e.g., ± 0.05 , ISO fit notations such as H7, or explicit upper/lower deviations).

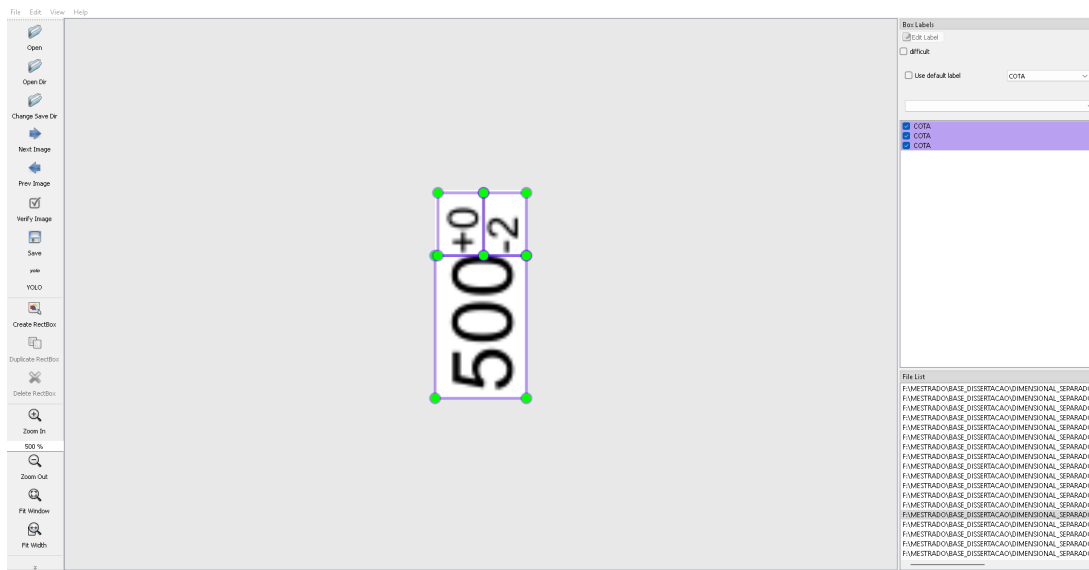


Figure 19 – Labeling Step IV: Detailed decomposition of nominal values and technical tolerances.

4.2 MODELS TRAINING

The core of the framework involves training a sequence of four distinct models, each responsible for a more granular level of drawing detail identification. Each model was trained using a balanced database and a cross-validation strategy to ensure its robustness.

4.2.1 Hyperparameter Optimization Strategy

To select the hyperparameter suitable values for model training, a stratified subset of the primary database was curated. This subset comprised 1,000 images —

500 from Class ‘3’ and 500 from Class ‘4’ — ensuring that the optimization process remained unbiased by the frequency distribution of the full dataset. This balanced approach allows the optimization algorithm to focus on feature extraction efficacy rather than class prevalence.

The tuning process targeted five critical hyperparameters: *number of epochs*, *batch size*, *initial learning rate*, *early stopping patience*, and the *optimizer type*. These suitable hyperparameters values were identified through the Optuna framework, an automated search paradigm that utilizes Bayesian optimization to explore the search space of possible values efficiently.

4.2.2 Optimization Setup and Search Space

The search space for the optimization was defined empirically, leveraging prior experimentation and established benchmarks in the literature. Table 4 summarizes the boundaries and categorical choices provided to the Optuna sampler. A total of 40 trials were executed to ensure a statistically significant exploration of the parameter landscape.

Table 4 – Optuna hyperparameter search space.

Hyperparameter	Search Space / Range	Distribution Type / Step
Epochs	[30, 100]	Int (step = 1)
Learning Rate (lr_0)	$[10^{-5}, 10^{-1}]$	Float (Logarithmic scale)
Batch Size	{16, 32, 64}	Categorical
Patience	[5, 20]	Int (step = 1)
Optimizer	{Adam, SGD, AdamW}	Categorical

For discrete parameters (*Epochs* and *Patience*), the search space uses an integer distribution (`suggest_int`) that tests values sequentially with a unitary step of 1. For the *Learning Rate*, a continuous logarithmic float distribution (`suggest_float` with `log=True`) is utilized, allowing the algorithm to automatically sample fine-grained values across non-linear magnitude scales.

4.2.3 Tuning Results and Sensitivity Analysis

The outcome of the 40-trial execution is visualized in Figure 20, which tracks the Mean Average Precision (mAP) across iterations. Interestingly, the optimal configuration was achieved in the very first trial, suggesting that the initial heuristic setup was highly aligned with the dataset’s characteristics. Nonetheless, the subsequent 39 trials provided the necessary data to validate this peak and analyze parameter sensitivity.

The computational cost of the hyperparameter optimization was quantified by analyzing the execution time of completed Optuna trials. In total, 40 trials successfully

completed, each requiring an average execution time of 341.94 minutes (approximately 5.7 hours), with a standard deviation of 208.67 minutes (approximately 3.5 hours), reflecting the variability inherent to the training process. Overall, the hyperparameter tuning procedure accumulated a total computational cost of approximately 227.96 hours (approximately 9.5 days), offering a transparent and reproducible estimate of the resources required for the proposed optimization strategy.

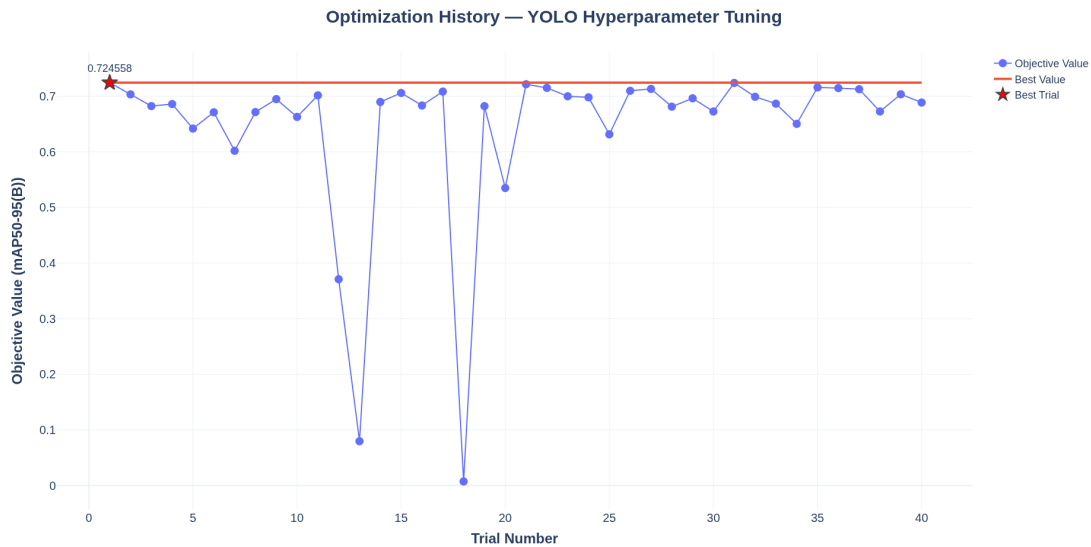


Figure 20 – Results of the 40-trial hyperparameter optimization study.

The best-performing configuration, which served as the baseline for the full-scale training of the four proposed models, is detailed in Table 5.

Table 5 – Optimal hyperparameters identified by the best-performing trial.

Parameter	Value
Best Trial Number	1
Max mAP	0.7246
Epochs	73
Batch Size	64
Initial Learning Rate (lr_0)	0.0058
Patience	15
Optimizer	AdamW

4.2.3.1 Parameter Importance Analysis

A critical insight from the Optuna study is the relative influence of each hyperparameter on model fitness. This importance was assessed using the fANOVA-based evaluator, as discussed in Section 2.1.5.2. The method estimates each parameter's contribution to the total variance of the objective function across the 40 completed trials. The resulting normalized scores, visualized in Figure 21, provide a clear hierarchy of the factors driving model performance.

The analysis reveals that **batch size** and **patience** are the most influential factors, collectively accounting for approximately 80% of the observed performance variance. This result highlights the dominant role of training stability and convergence control in the optimization of the proposed CNN models. Conversely, the choice of *optimizer* contributed less than 1% to the overall performance, indicating that for this specific industrial dataset, the specific variant of the gradient descent algorithm (e.g., AdamW vs. SGD) is less critical than the structural training regime.

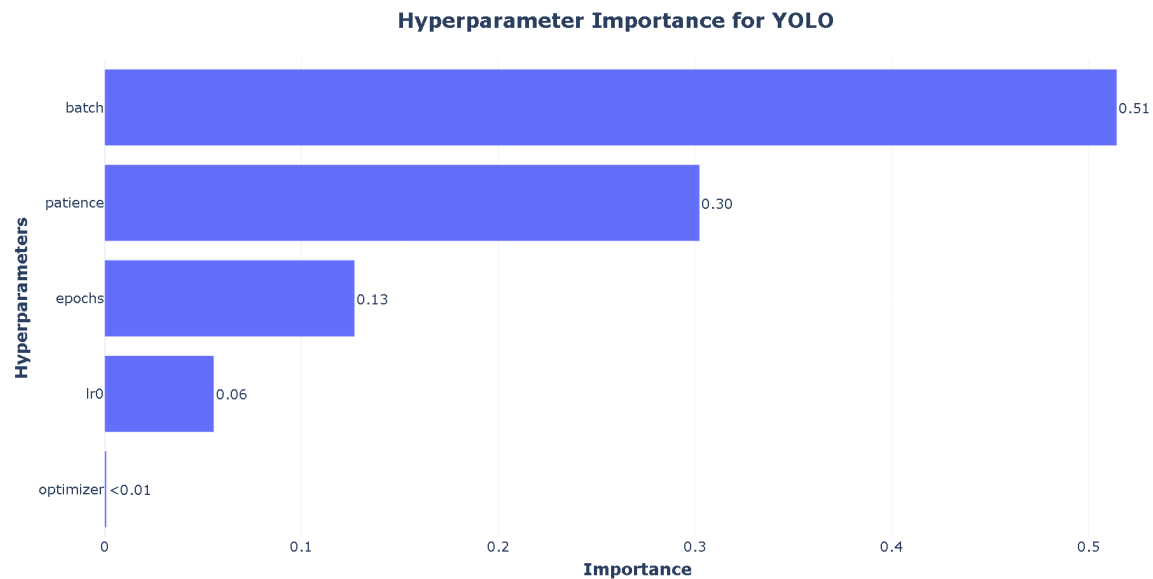


Figure 21 – Hyperparameter importance plot generated via fANOVA, illustrating the dominance of batch size and patience.

Global Sensitivity Analysis via SHAP

To complement the fANOVA results and understand the directional impact of each parameter, a global SHAP (SHapley Additive exPlanations) analysis was conducted (Figure 22). While fANOVA identifies which parameters are influential, SHAP clarifies how different values of those parameters push the model's performance (mAP) up or down.

Although the SHAP summary plot indicates that larger batch sizes tend to negatively impact performance on average, the best-performing trial utilized a **batch size of 64**. This apparent contradiction highlights the existence of local optima in the multi-dimensional hyperparameter space. In this specific configuration, the potential adverse effects of a larger batch size were mitigated by a synergistic combination of *learning rate* and *patience*, resulting in a highly stable and high-performing training regime. This observation underscores the importance of considering hyperparameter interactions rather than interpreting individual parameters in isolation.

Visualization and Discretization Strategy

Hyperparameters in the SHAP summary plot are ordered according to their mean absolute SHAP values, ensuring consistency with standard interpretability conventions. To facilitate the analysis of continuous ranges, values were discretized into low, medium, and high regimes based on feature-wise quantiles. This representation highlights that intermediate hyperparameter values often exert a meaningful influence on model performance, rather than acting as neutral transition points.

For categorical variables, such as the *optimizer*, the values were mapped to an ordinal scale (Adam, SGD, AdamW) to allow for comparative visualization. This ordering is used solely for interpretability purposes and does not imply an intrinsic hierarchy among the algorithms.

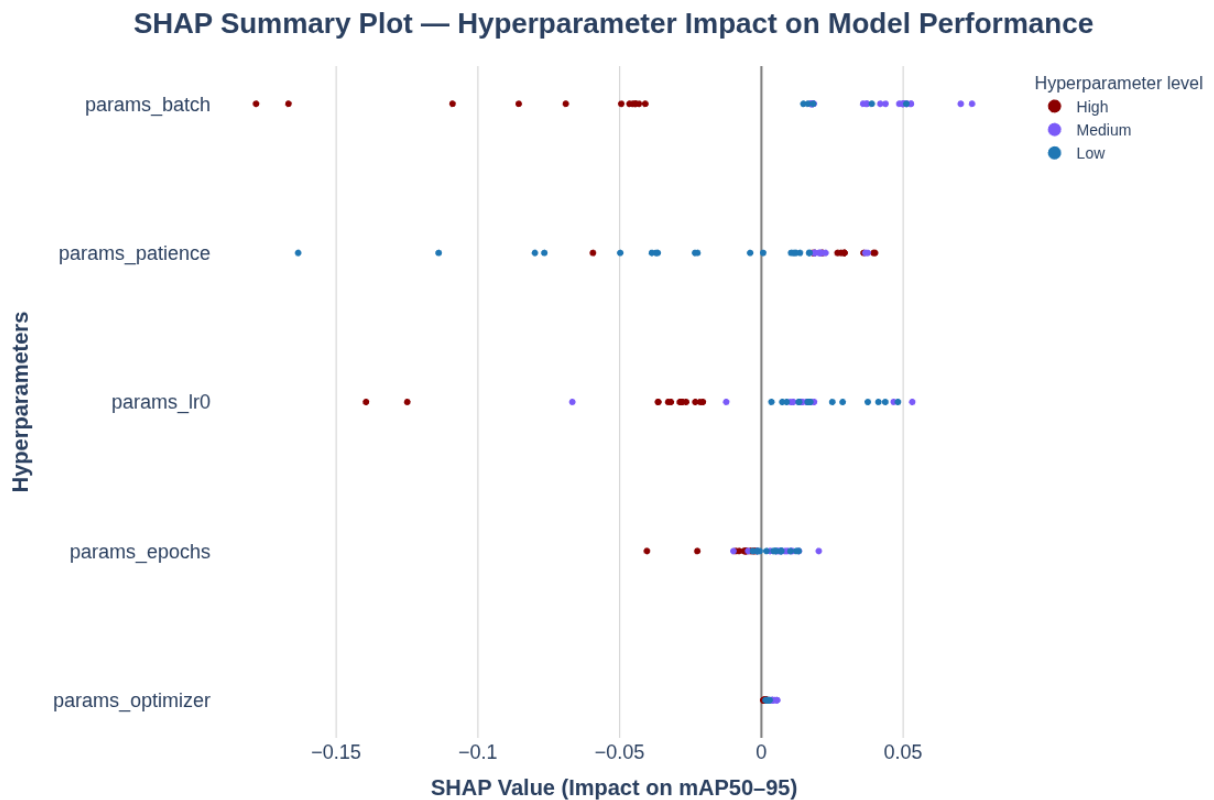


Figure 22 – SHAP summary plot illustrating the directional influence of hyperparameters on the fitness score (mAP).

4.2.4 K-Fold Cross-Validation

Following the hyperparameter optimization, each model in the pipeline was trained and evaluated independently. A 5-fold cross-validation scheme ($K = 5$) was employed to ensure robustness and provide reliable validation metrics, with the choice of K defined empirically.

Model Selection Strategy

Beyond performance estimation, the cross-validation process serves as a framework for **Model Selection**. In this study, following the K iterations, the model weights from the fold that achieved the highest performance on its respective validation set were selected as the final candidate for the inference pipeline. This approach is justified by the need to identify the specific model instance that demonstrated the best generalization capability across the diverse industrial samples encountered during the K trials (RASCHKA, 2018; HASTIE; TIBSHIRANI; FRIEDMAN, 2009).

4.2.5 Model 1: Machine View Detection

The first model of the pipeline, responsible for localizing the primary machine views, was evaluated through a rigorous two-step validation process: internal cross-validation and external temporal validation.

4.2.5.1 Training and Cross-Validation Analysis

To mitigate bias and ensure stability, Model 1 was trained on a balanced subset of 2,000 images (1,000 from Class ‘3’ and 1,000 from Class ‘4’). The training was conducted utilizing the optimal hyperparameters identified via the Optuna study (see section 4.2.3).

The results of the 5-fold cross-validation are summarized in Table 6. The performance remained remarkably consistent across all folds, with the mean mAP calculated at 0.7138 ± 0.0029 . This low standard deviation (less than 0.5%) indicates that the model is highly stable and that its ability to generalize is independent of the specific data split used for training.

Table 6 – K-Fold Cross-Validation Summary - Model 1 (Global View Detector).

Fold	Best mAP	Best Epoch	Rank	Processing Time (h)
Fold 4	0.71797	50	1	12.22442
Fold 3	0.71570	56	2	12.77914
Fold 2	0.71246	37	3	9.79953
Fold 5	0.71187	46	4	10.51250
Fold 1	0.71112	69	5	13.32347
Mean $\pm$ Std Dev	0.7138 $\pm$ 0.0029	-	-	11.72781 $\pm$ 1.507754

The convergence dynamics of the training process are illustrated in Figures 23, 24, and 25, which present the Box, Classification, and Distribution Focal Loss (DFL) trajectories across the five data partitions. The training curves exhibit a monotonic and

consistent decline across all three metrics, confirming that the architecture efficiently learned the spatial features required for view detection without early stagnation.

In contrast, the validation curves exhibit the expected high-frequency oscillations characteristic of complex bounding box regression tasks during the initial learning phases. However, global stabilization is achieved across all folds after approximately 40 epochs. Fold 4 (highlighted in red) was selected as the optimal model instance; despite experiencing transient spikes during validation, it ultimately demonstrated the most favorable balance between convergence and generalization, securing the highest individual mAP of 0.718.

To provide a holistic view of the architecture's stability, a mean loss trajectory (depicted by the thick black line) was computed by averaging the epoch-wise loss values across all five folds. While this composite curve does not represent a deployable model instance, it serves as a critical diagnostic tool, visually confirming that the underlying learning tendency achieves a stable horizontal plateau, proving mathematical convergence resilient to specific data splits.

The mean loss terminates prematurely because its calculation depends concurrently on all five folds; therefore, as soon as the first individual fold triggers the early stopping criterion and concludes its execution, the composite average is discontinued.

Furthermore, the vertical axes (Y-axis) for the training and validation loss trajectories are scaled independently across the evaluations. The validation plots inherently display higher initialization bounds and stochastic spikes. As learning stabilizes, these curves converge asymptotically. This standardized visual behavior, scale discrepancy, and stopping mechanics apply uniformly to all four specialized models within the proposed cascade pipeline, ensuring a consistent and rigorous evaluation paradigm across the entire architectural framework.

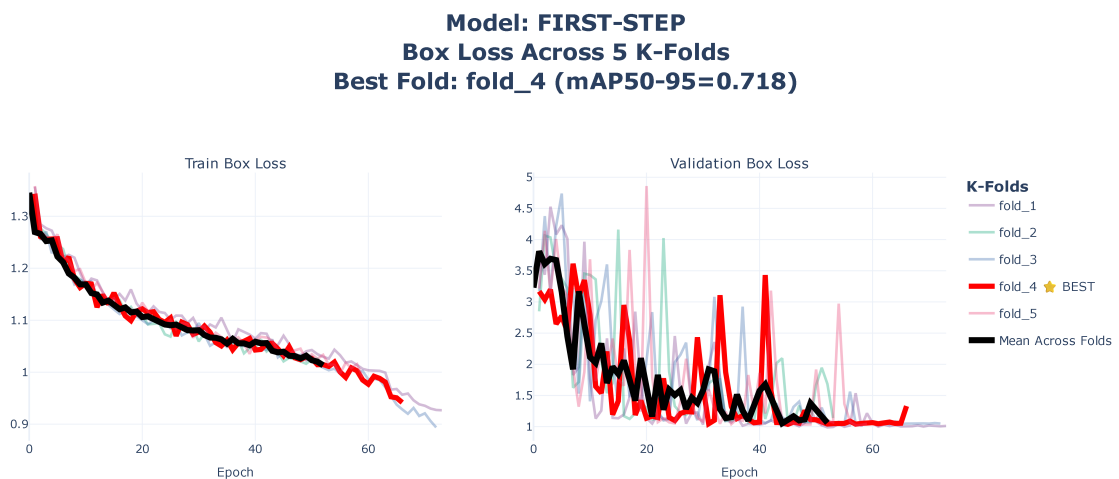


Figure 23 – Training and Validation Box Losses Across 5 K-Folds - Model 1

Model: FIRST-STEP
Classification Loss Across 5 K-Folds
Best Fold: fold_4 (mAP50-95=0.718)

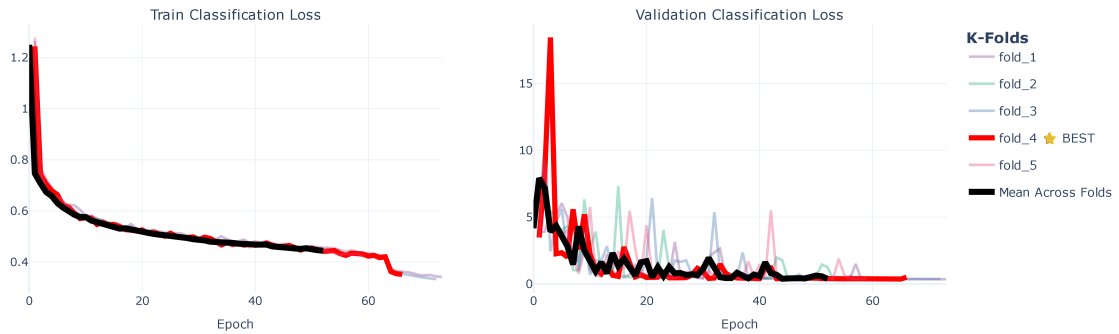


Figure 24 – Training and Validation Classification Losses Across 5 K-Folds - Model 1

Model: FIRST-STEP
DFL Loss Across 5 K-Folds
Best Fold: fold_4 (mAP50-95=0.718)

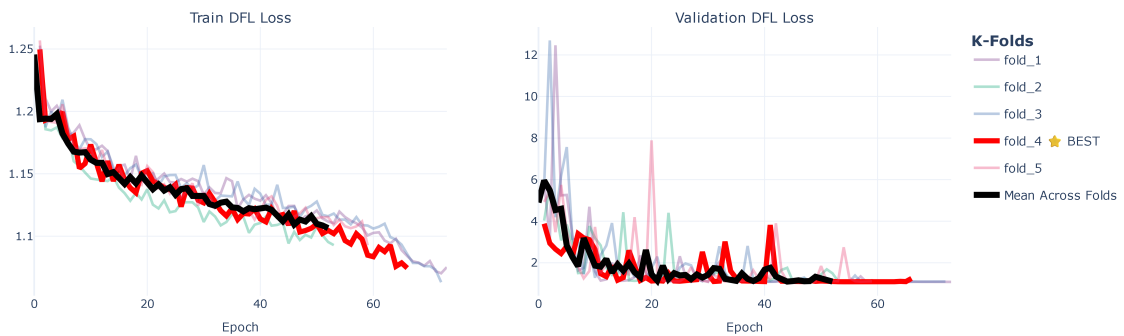


Figure 25 – Training and Validation DFL Losses Across 5 K-Folds - Model 1

4.2.5.2 Temporal Out-of-Sample Validation

Following the training phase, the weights from Fold 4 (the best-performing instance) were selected for evaluation against a dedicated out-of-sample dataset. This dataset consisted of 77 documents generated during the year 2025. This temporal separation is crucial as it ensures the evaluation reflects the model's reliability in a real-world production environment, where drafting styles and machine revisions may have evolved compared to the historical training data.

As shown in Figure 26, the model achieved a Precision of 1.000, meaning no false positives were recorded, a vital requirement for industrial automation to avoid incorrect data entry. The Recall of 0.984 indicates that only a negligible fraction of views was missed. Furthermore, the mAP@0.5 reached 0.995, and the Fitness score of 0.834 confirms the robustness of the detector.

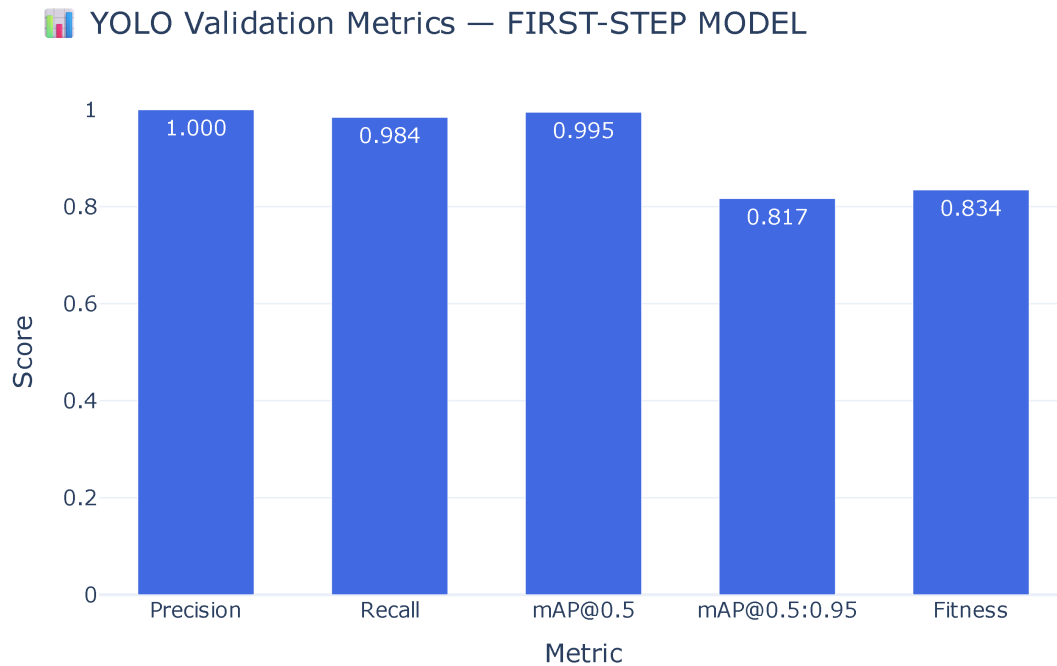


Figure 26 – Validation metrics for Model 1 (Machine View Detection) on the 2025 temporal validation set.

These metrics substantiate that Model 1 provides a dependable foundation for the subsequent layers of the pipeline. By successfully isolating the views with near-perfect precision, it ensures that the downstream models (Orientation and Dimension extraction) receive high-quality, relevant visual inputs.

4.2.6 Model 2: Machine View Type Classification

The second model of the pipeline focuses on the semantic identification of the isolated machine views. This classification is vital for establishing the spatial context required by the subsequent Multimodal Large Language Model (MLLM) reasoning stage (Stage 3 from Figure 6).

4.2.6.1 Training and Cross-Validation Analysis

For this model, the regions identified by the initial detector (Model 1) were labeled according to their orthographic perspective: Front View (*VF*), Side View (*VL*), and Top View (*VS*). As shown in Table 7, a balanced dataset consisting of 1,527 samples (509 per class) was utilized for training. This perfect balance ensures that the model does not develop a bias toward the most frequent drawing layouts (e.g., standard three-view drawings).

The training followed the same 5-fold cross-validation protocol on the high-performance workstation. The results, summarized in Table 8, demonstrate exceptional

Table 7 – Label distribution for orthographic machine view classes.

Class ID	Label	Sample Count
0	Front View (VF)	509
1	Side View (VL)	509
2	Top View (VS)	509

stability. The mean mAP achieved was 0.9797 ± 0.0081 . Even with a slight performance drop in Fold 4, the overall standard deviation remains below 1%, confirming that the YOLOv8 architecture can reliably distinguish between complex mechanical perspectives across different data partitions.

Table 8 – K-Fold Cross-Validation Summary - Model 2 (Orientation Classifier).

Fold	Best mAP	Best Epoch	Rank	Processing Time (h)
Fold 2	0.98533	72	1	5.87675
Fold 3	0.98479	69	2	5.59131
Fold 1	0.98334	67	3	6.06617
Fold 5	0.97905	52	4	5.88672
Fold 4	0.96580	40	5	4.34083
Mean $\pm$ Std Dev	0.97966 $\pm$ 0.00813	-	-	5.55236 $\pm$ 0.69827

To evaluate the learning efficiency of the orientation classification task, the Box, Classification, and DFL loss metrics were tracked across all five data partitions, as presented in Figures 27, 28, and 29. The training curves demonstrate a steady, uniform decay, indicating that the network rapidly acquired the discriminative features necessary to differentiate between the distinct orthographic perspectives.

In this specific classification context, the validation curves exhibit markedly less high-frequency oscillation compared to the initial view detection model. Global stabilization is achieved much earlier in the training cycle, plateauing around epoch 20. Fold 2 (highlighted in red) emerged as the most robust partition, achieving an exceptional mAP of 0.985. Furthermore, the tight clustering of all individual validation folds around the composite mean trajectory (black line) underscores the high reliability of the classifier, proving its strong resilience against variance in the training data splits.

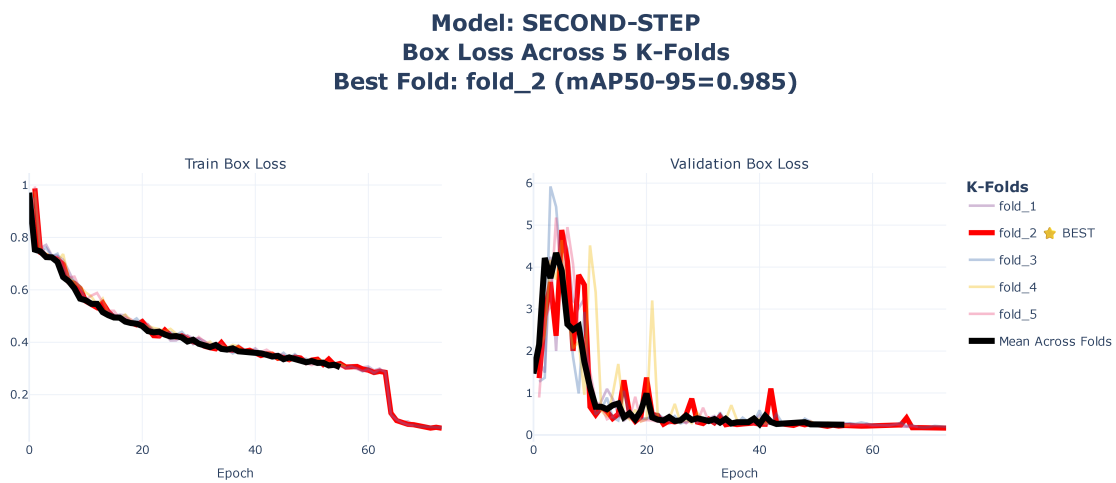


Figure 27 – Training and Validation Box Losses Across 5 K-Folds - Model 2

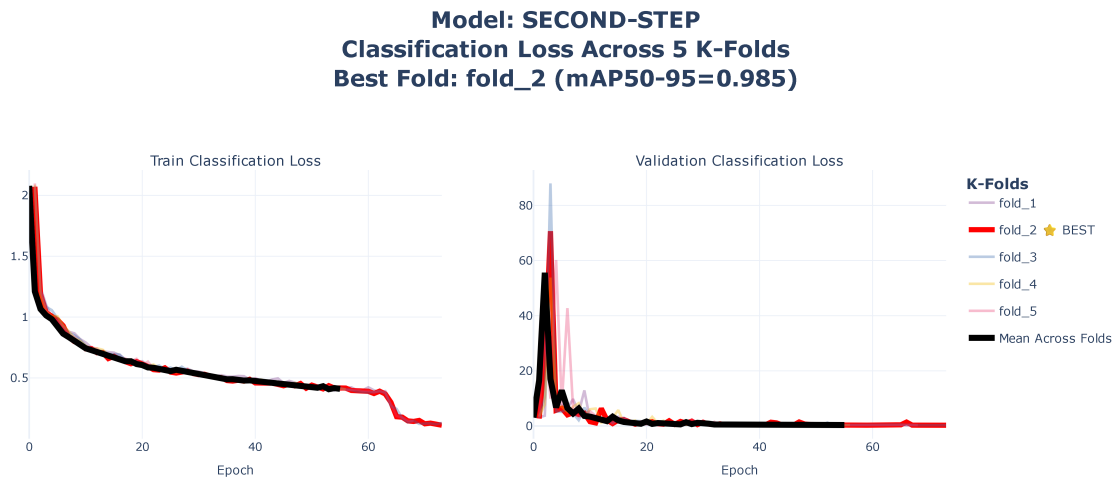


Figure 28 – Training and Validation Classification Losses Across 5 K-Folds - Model 2

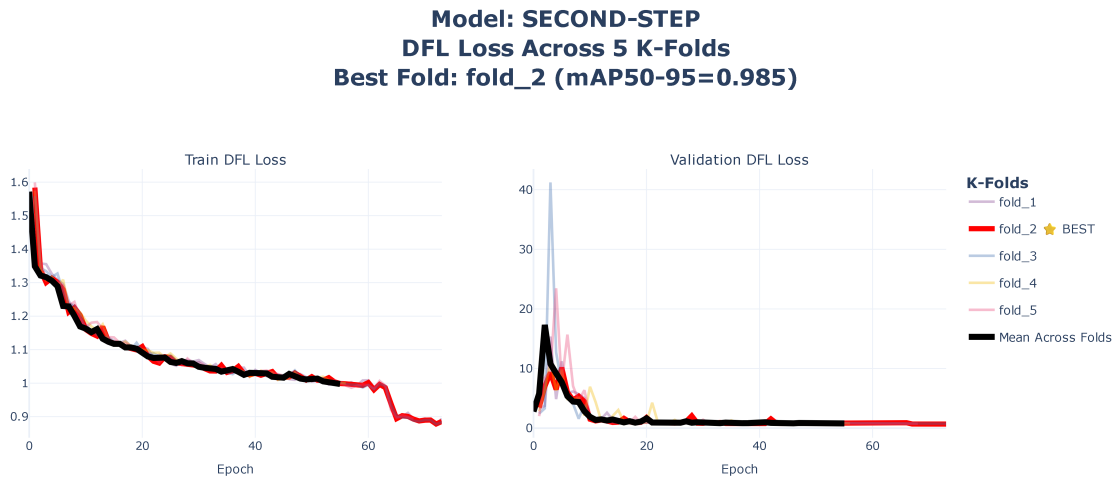


Figure 29 – Training and Validation DFL Losses Across 5 K-Folds - Model 2

4.2.6.2 Temporal Out-of-Sample Evaluation

To validate the model's robustness in a production-ready environment, the weights from Fold 2 were evaluated against a temporal hold-out set. This set comprised 247 views extracted from drawings generated in 2025, representing the current drafting standards of the organization.

The validation metrics, illustrated in Figure 30, confirm the high fidelity of the classifier. The model achieved a Precision of 0.987 and a Recall of 0.930. Notably, the mAP@0.5 and mAP@0.5:0.95 scores are nearly identical, 0.944 and 0.942, respectively, which indicates that the model is extremely precise in its bounding box localization, even at very strict Intersection over Union (IoU) thresholds.

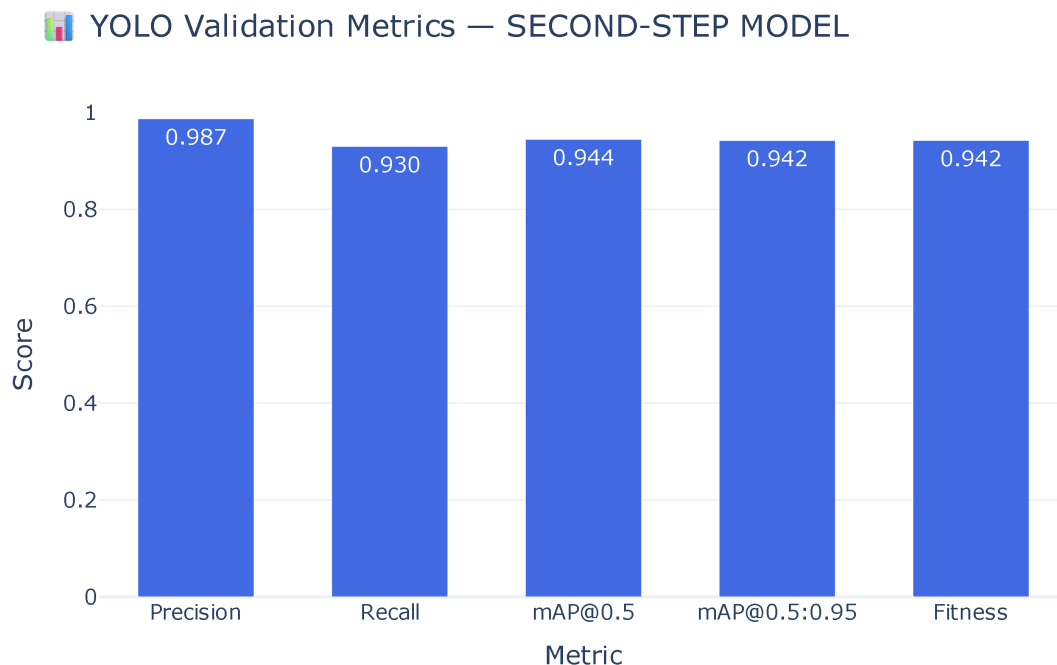


Figure 30 – Validation metrics for Model 2 (Orientation Classifier) on the 2025 out-of-sample set.

With a final fitness score of 0.942, Model 2 successfully automates the spatial orientation task, providing the necessary “geometric anchor” for the pipeline’s subsequent extraction of technical dimensions.

4.2.7 Model 3: Machine Dimension Detection

The third model of the proposed architecture moves from view-level classification to the localization of technical annotations. This model is responsible for identifying and bounding all dimensional information within the previously isolated machine views, effectively separating semantic text from geometric primitives.

4.2.7.1 Training and Cross-Validation Analysis

A specialized balanced database was constructed for this task, utilizing 2,500 high-resolution machine views where all dimension blocks were manually annotated by specialists. The training process followed the established K -fold cross-validation protocol using the optimal hyperparameters previously identified.

The cross-validation results, detailed in Table 9, show high stability across all partitions. The mean mAP achieved was 0.9209 ± 0.0044 . Despite the increased visual complexity of detecting multiple small objects within a single image, the low standard deviation confirms the model's robustness and its consistent ability to generalize across different styles of technical drafting.

Table 9 – K-Fold Cross-Validation Summary - Model 3 (Dimension Block Locator).

Fold	Best mAP	Best Epoch	Rank	Processing Time (h)
Fold 3	0.92625	72	1	10.08514
Fold 5	0.92266	69	2	10.30072
Fold 2	0.92144	68	3	10.17347
Fold 1	0.92006	70	4	10.33681
Fold 4	0.91407	69	5	10.18903
Mean $\pm$ Std Dev	0.9209 $\pm$ 0.0044	-	-	10.21703 $\pm$ 0.10177

The training progression for the dimension block localization task is captured in the loss metrics presented in Figures 31, 32, and 33. The training loss trajectories display a continuous, stable descent across all parameters (Box, Classification, and DFL). Despite the increased visual complexity associated with detecting multiple, high-density small annotations within a single view, the network efficiently minimizes both spatial and categorical prediction errors without stagnation.

The validation curves reflect a rapid architectural adaptation, reaching a steady-state plateau by approximately epoch 20 with remarkably minimal oscillation. Fold 3 (highlighted in red) was identified as the optimal partition, yielding the highest individual mAP of 0.926. While this specific fold experienced a brief transient spike in validation classification loss near epoch 35, it quickly recovered, confirming that the model's overall generalization capabilities remained uncompromised. Furthermore, the exceptionally tight alignment between the individual data splits and the aggregate mean trajectory (black line) underscores a high degree of consistency, proving the pipeline's reliability in isolating critical dimension blocks regardless of the specific training subset.

Model: THIRD-STEP
Box Loss Across 5 K-Folds
Best Fold: fold_3 (mAP50-95=0.926)

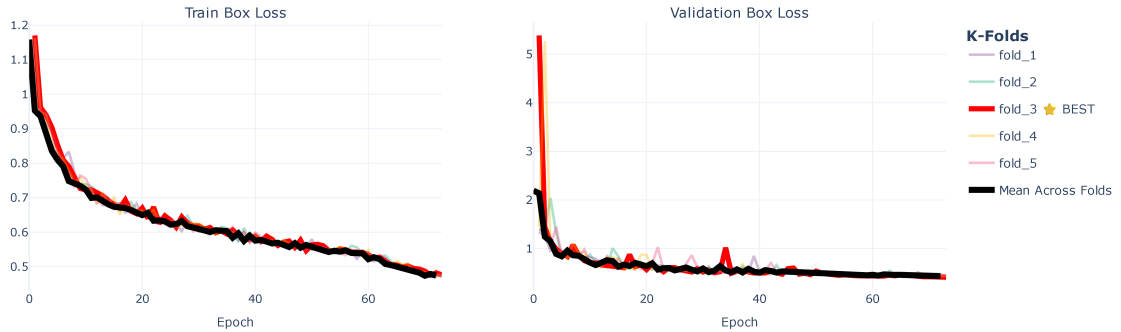


Figure 31 – Training and Validation Box Losses Across 5 K-Folds - Model 3

Model: THIRD-STEP
Classification Loss Across 5 K-Folds
Best Fold: fold_3 (mAP50-95=0.926)

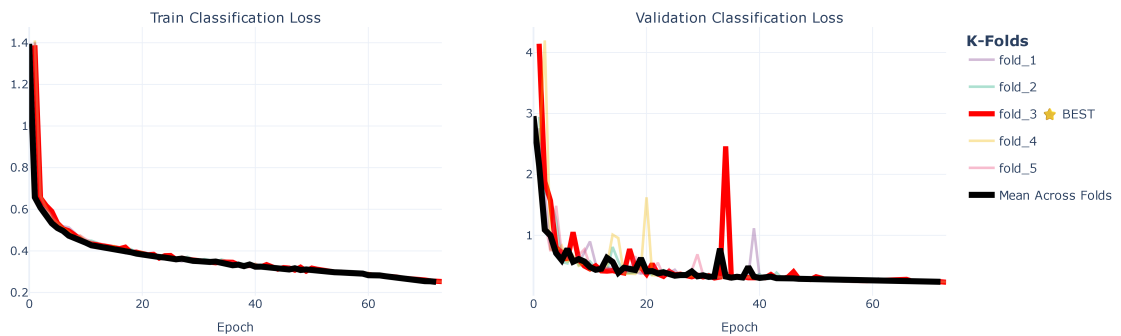


Figure 32 – Training and Validation Classification Losses Across 5 K-Folds - Model 3

Model: THIRD-STEP
DFL Loss Across 5 K-Folds
Best Fold: fold_3 (mAP50-95=0.926)

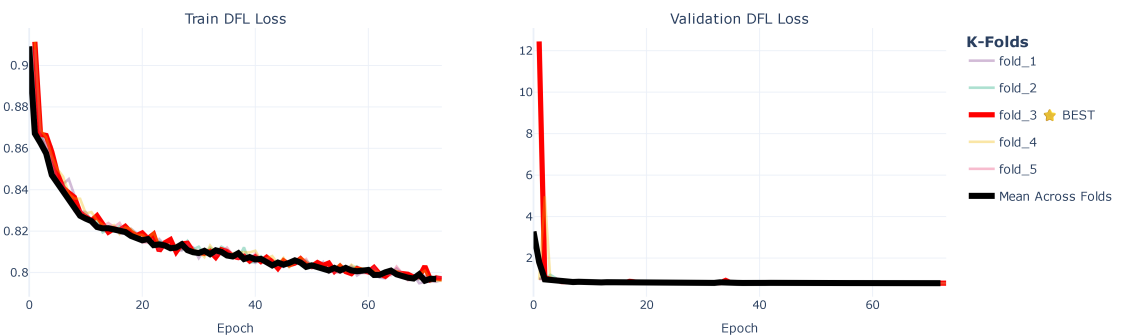


Figure 33 – Training and Validation DFL Losses Across 5 K-Folds - Model 3

4.2.7.2 Temporal Out-of-Sample Evaluation

To ensure the system's readiness for current industrial standards, the weights from Fold 3 were evaluated against a temporal out-of-sample dataset. This set comprised 250 views extracted from documents generated in 2025, providing a clear benchmark of the model's performance on contemporary documentation.

The evaluation metrics, shown in Figure 34, reveal a highly reliable detection profile. The model achieved a Precision of 0.983 and a Recall of 0.975. The near-perfect mAP@0.5 of 0.991 indicates that the model almost never fails to identify a dimension block. Even at the stricter mAP@0.5:0.95 threshold, the model maintained a score of 0.915, proving that the bounding boxes are tightly and accurately fitted around the alphanumeric data.

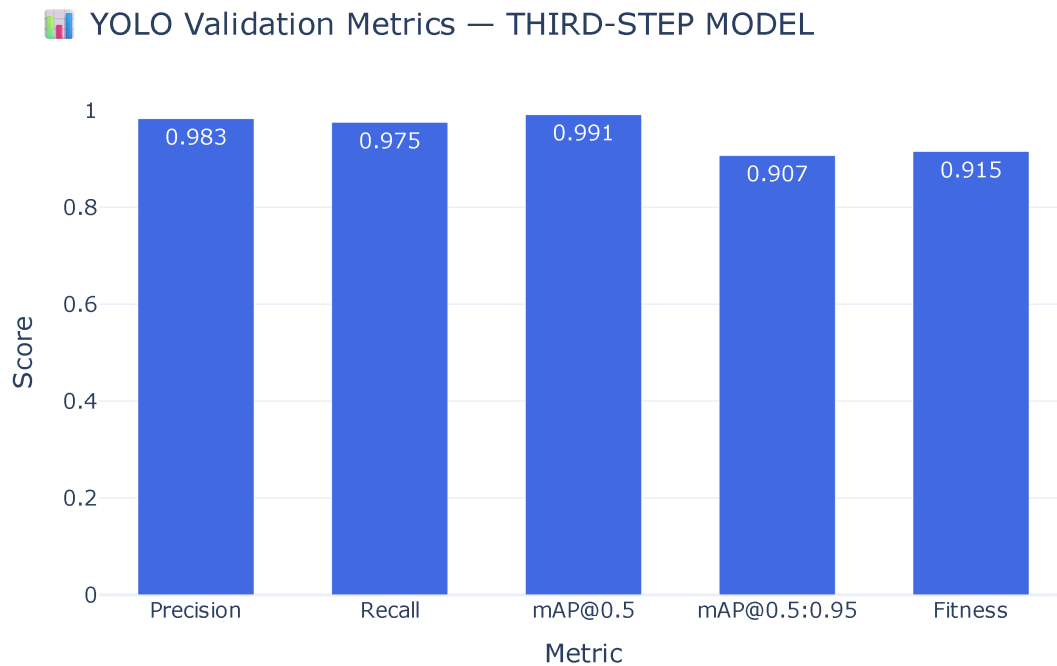


Figure 34 – Validation results for Model 3 (Dimension Block Locator) on the 2025 temporal validation set.

The resulting fitness score of 0.923 reinforces that Model 3 is suitable for the pipeline. By providing clean, localized crops of the dimensional data, it eliminates background geometric noise, allowing the final model in the chain to focus exclusively on the internal structure of the dimension (value vs. tolerance).

4.2.8 Model 4: Dimension and Tolerance Classification

The final model of the detection pipeline is responsible for the granular classification of dimension blocks. This model distinguishes between nominal dimensions

and those that include technical tolerances, a distinction that is fundamental for the downstream logic of automated verification.

4.2.8.1 Training and Cross-Validation Analysis

For this task, the dataset identified in the previous step was refined to create a perfectly balanced training environment. The database comprised 3,938 samples, split equally into two classes: dimensions **without tolerances** (nominal only) and dimensions **with tolerances** (nominal plus deviation).

The training utilized the established K -fold cross-validation scheme. The results, summarized in Table 10, demonstrate a mean mAP of 0.9431 ± 0.0036 . The consistently low standard deviation confirms that the YOLOv8 architecture, once optimized, becomes highly expert in recognizing the small typographical and structural patterns that characterize engineering tolerances, regardless of the document's origin.

Table 10 – K-Fold Cross-Validation Summary - Model 4 (Tolerance Classifier).

Fold	Best mAP	Best Epoch	Rank	Processing Time (h)
Fold 2	0.94681	72	1	14.32308
Fold 3	0.94531	69	2	14.66522
Fold 4	0.94416	68	3	13.79456
Fold 5	0.94251	70	4	12.28222
Fold 1	0.93649	69	5	14.10719
Mean $\pm$ Std Dev	0.9431 $\pm$ 0.0040	-	-	13.83446 $\pm$ 0.92390

To assess the fine-grained discrimination capabilities of the final detection model, the learning dynamics for the Box, Classification, and DFL losses are detailed in Figures 35, 36, and 37. The training trajectories exhibit a tightly clustered, uniform exponential decay across all metrics. This indicates that the network efficiently extracted the subtle typographical and structural features required to distinguish a nominal dimension from its respective technical tolerance.

In the validation phase, the network experiences sharp initial volatility within the first 10 epochs, a common behavior when optimizing for highly localized, minute graphical differences. However, this early turbulence is rapidly overcome, transitioning into a remarkably flat and stable plateau by epoch 20. Fold 2 (highlighted in red) emerged as the optimal candidate, securing a superior mAP of 0.947. The strict convergence of all individual data partitions toward the composite mean trajectory (black line) in the later epochs acts as a strong empirical proof of the model's reliability, confirming its robustness against diverse industrial annotation styles.

Model: FOURTH-STEP
Box Loss Across 5 K-Folds
Best Fold: fold_2 (mAP50-95=0.947)

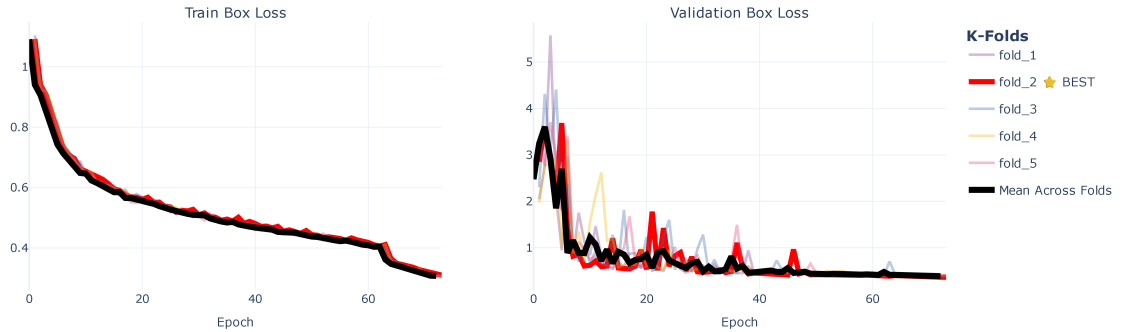


Figure 35 – Training and Validation Box Losses Across 5 K-Folds - Model 4

Model: FOURTH-STEP
Classification Loss Across 5 K-Folds
Best Fold: fold_2 (mAP50-95=0.947)

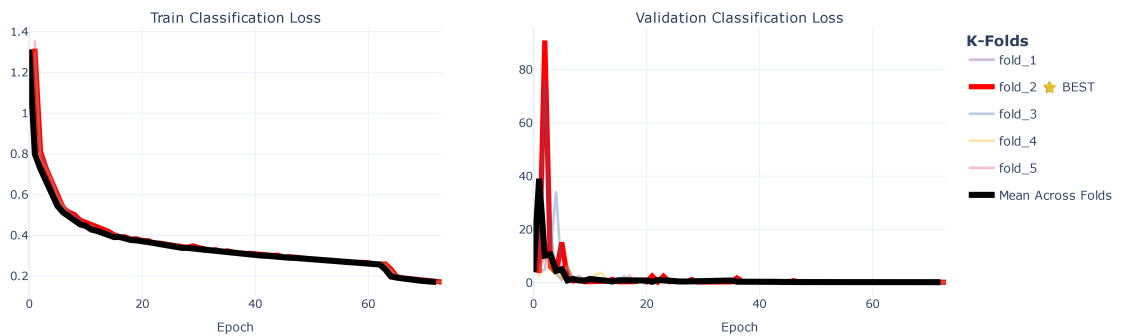


Figure 36 – Training and Validation Classification Losses Across 5 K-Folds - Model 4

Model: FOURTH-STEP
DFL Loss Across 5 K-Folds
Best Fold: fold_2 (mAP50-95=0.947)

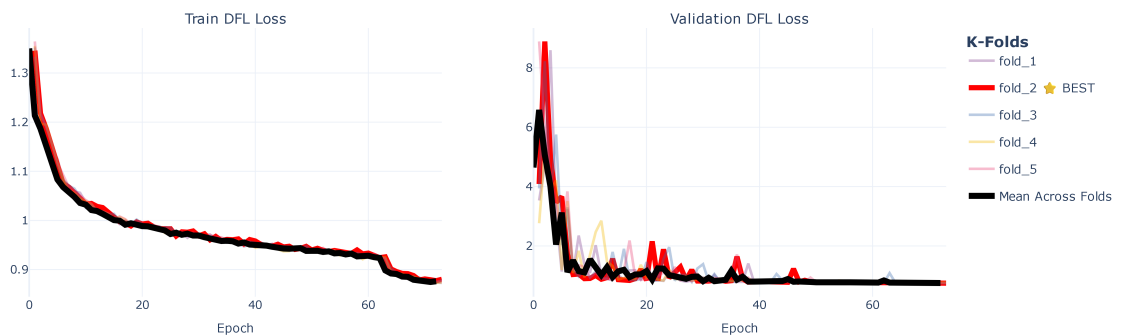


Figure 37 – Training and Validation DFL Losses Across 5 K-Folds - Model 4

4.2.8.2 Temporal Out-of-Sample Evaluation

To conclude the validation of the detection layers, the weights from Fold 2 were tested against an extensive temporal out-of-sample dataset. This set included 2,923 dimension images extracted from drawings generated in 2025.

The performance metrics, as shown in Figure 38, represent the highest accuracy achieved in the pipeline. The model reached a Precision of 0.986 and a Recall of 0.995, virtually eliminating missed detections. Notably, the mAP@0.5:0.95 of 0.967 is consistently high for industrial data, ensuring that the model's bounding boxes are localized with suitable precision around the tolerance strings.

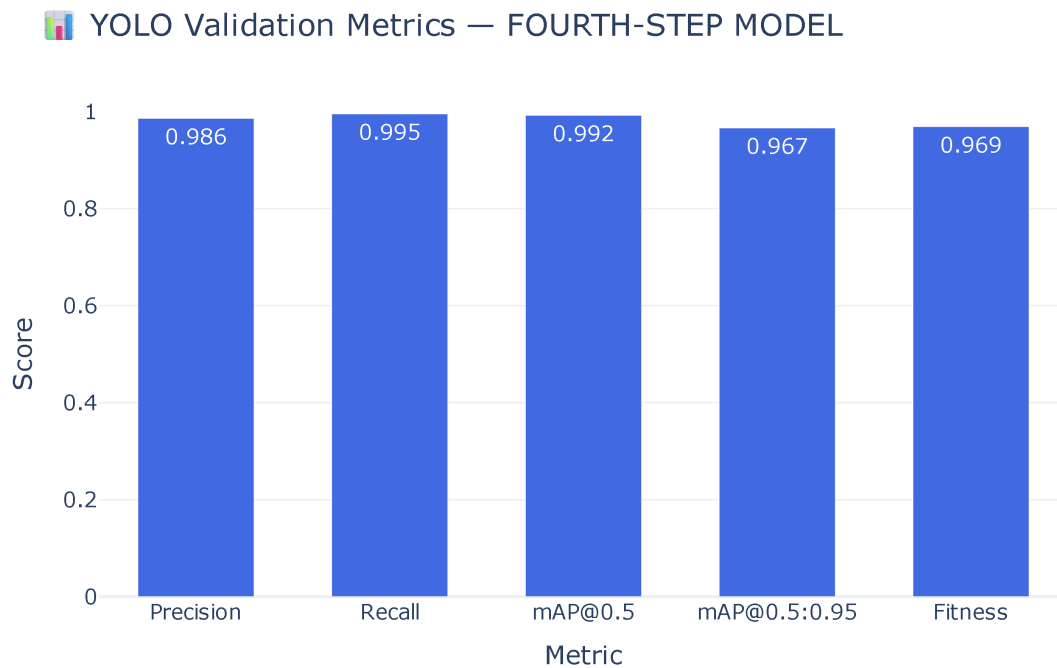


Figure 38 – Validation results for Model 4 (Tolerance Classifier) using 2,923 labeled images from 2025.

4.3 MULTIMODAL REASONING AND SEMANTIC VERIFICATION

While the YOLOv8 pipeline excels at spatial localization and classification, the final extraction of engineering intent requires a semantic understanding of technical drawing conventions. To address this, the Gemini 2.5 Pro Multimodal Large Language Model (MLLM) was integrated into the workflow (COMANICI et al., 2025).

This model was selected due to its state-of-the-art performance in *Vision-Language* tasks and its expansive 1-million-token context window. These features are essential for processing high-resolution visual crops alongside extensive technical definitions and cross-referencing rules. To comply with corporate data protection policies and safeguard proprietary intellectual property, the model was accessed via a secure,

private instance, ensuring that no sensitive engineering data was exposed to public training sets.

4.3.1 Prompt Engineering and Instruction Design

The interaction with the MLLM was governed by a meticulously structured system prompt, designed to minimize hallucinations and ensure output consistency. The prompt architecture followed a multi-layered design, as detailed below:

- **Context and Role:** Defines the MLLM’s persona as an expert industrial engineering auditor.
- **Descriptive Objective:** Clearly states the goal of extracting and validating dimensions against the provided definitions.
- **Task Specification:** Instructions for analyzing the relationship between the nominal value and its tolerance within the specific machine view.
- **Data and Output Format:** Strict requirement for structured data delivery (e.g., JSON), facilitating the automated integration with the Production System.
- **Few-Shot Learning:** Inclusion of output examples to guide the model’s reasoning pattern.
- **Constraints and Rules:** Explicit “guardrails” to prevent the interpretation of geometric noise as technical data.
- **Final Observations:** Specific guidance on handling ambiguous annotations or low-contrast legacy text.

The complete system prompt utilized in this study is provided in Appendix A.1.

4.3.2 Multimodal Input Pipeline

For each inference cycle, the MLLM receives a composite input consisting of the structured prompt (the “Question”) and the localized image crops generated by the YOLOv8 pipeline (Figure 10). To provide the model with the necessary “ground truth” for verification, a set of measure definitions and technical standards (Appendix A.2) is injected into the context window.

This multimodal fusion allows the Gemini 2.5 Pro to not only “read” the text via its internal OCR capabilities but also to “reason” about the spatial orientation of the dimension relative to the machine’s view type (Front, Side, or Top), as identified by the earlier models.

4.4 EVALUATION PIPELINE

To validate the proposed framework, a comprehensive evaluation approach was conducted using real-world engineering data. The evaluation pipeline focused on assessing the workflow ability to correctly identify, localize, and verify dimensional data against the company's database.

4.4.1 Dataset Composition

The evaluation dataset consisted of 77 engineering documents generated during the year 2025. To maintain methodological consistency, these are the exact same out-of-sample documents utilized during the temporal validation of the initial View Detection model (Section 4.2.5). These documents were sampled from the company's recent archives to ensure the evaluation reflected current drafting standards and electric machine complexities.

- **Sample Size:** 77 unique engineering drawings (in .pdf format) from the 2025 production cycle.
- **Evaluation Iterations:** To ensure statistical robustness and stability of the results, the entire evaluation set was processed through 20 independent iterations.
- **Total Processed Instances:** This resulted in a cumulative sample of 1,540 processed documents (77×20).
- **Dimensional Density:** On average, each document contained approximately **10 critical dimensions** requiring verification.
- **Total Data Points:** This yielded a comprehensive evaluation set of approximately 15,400 individual dimensional verification tasks, providing a high degree of confidence in the reported metrics.

4.4.2 Evaluation Tasks: Measures vs. Dimensions

To provide a granular analysis of the pipeline's performance, the evaluation was bifurcated into two distinct tasks: the semantic identification of the variable (*Measure*) and the extraction of its numerical content (*Dimension*).

- **Measures (Semantic Identification):** This task evaluates the workflow's ability to correctly associate a bounding box with its specific engineering variable name (e.g., identifying that a specific location on the drawing corresponds to the variable 'AD', 'AB', or 'L'). Success here indicates correct semantic reasoning and spatial localization.

- **Dimensions (Value Extraction):** This task evaluates the workflow's ability to accurately read the alphanumeric value associated with that measure (e.g., extracting the text '30', '50 ± 0.1', or '100'). Success here indicates correct OCR performance and text parsing.

This distinction allows for identifying whether errors stem from a failure to understand where a value is (Measure error) or a failure to read what the value is (Dimension error).

4.5 CHAPTER CONSIDERATIONS

This chapter presents the complete methodology applied to each model, including database preparation, model training and evaluation procedures, as well as the evaluation pipeline used to test out-of-sample documents.

5 RESULTS

This chapter presents the quantitative and qualitative evaluation of the proposed end-to-end automated extraction framework. Section 5.1 details the overall system performance, analyzing precision, recall, and stability across multiple inference trials. Section 5.2 provides a comprehensive analysis of the primary challenges and edge cases encountered, discussing the inherent limitations of integrating Computer Vision with MLLM reasoning in complex industrial scenarios.

5.1 END-TO-END SYSTEM PIPELINE RESULTS

To approximate real-world operational conditions, the fully integrated pipeline was evaluated on the temporal hold-out test set. The system was required to identify ten distinct measurement categories (e.g., AD1, AD, AL, A), each representing a specific type of dimensional or annotation pattern present in legacy technical drawings. Table 11 summarizes the final performance metrics for the end-to-end system, distinguishing between two critical evaluation categories: **Measures** and **Dimensions**. In this context, **Measures** refers to the semantic identification of geometric parameters (e.g., correctly classifying a dimension as ‘AB’ or ‘AD’ based on its position), whereas **Dimensions** refers to the accurate extraction of the corresponding numerical values (e.g., correctly retrieving the text ‘30’).

Table 11 – Performance Metrics for Measures and Dimensions

Metric	Measures	Dimensions
Precision	0.88258 ± 0.00330	0.93030 ± 0.00442
Recall	0.93930 ± 0.0843	0.94220 ± 0.00802
F1-Score	0.91005 ± 0.00536	0.93620 ± 0.0542
IoU	0.83498 ± 0.00898	0.88009 ± 0.00954

The quantitative results presented in Table 11 demonstrate that the end-to-end system exhibits strong performance across both evaluated categories—Measures and Dimensions. For Measures, the precision (0.88258), recall (0.93930), and F1-score (0.91005) indicate a balanced detection capability, with the model achieving comparable rates of true-positive identification and false-positive suppression. The Jaccard Index (IoU) of 0.83498 further corroborates this stability, confirming a high degree of overlap between the set of predicted detections and the ground truth, consistent with the strong F1 performance.

Similarly, the performance on Dimensions is consistently higher across all metrics. The system achieves a precision of 0.93030 and an F1-score of 0.93620,

reflecting its strong reliability in correctly identifying dimension annotations with minimal false positives. The Jaccard Index (IoU) of 0.88009 further validates this performance, indicating a substantial overlap between the set of detected dimensions and the ground truth, reinforcing the high classification accuracy. The recall value (0.94220), while high, is comparatively lower than the corresponding precision, suggesting that the model adopts a conservative detection strategy that occasionally results in missed dimension annotations.

Although the results for *Dimensions* are numerically superior to those obtained for *Measures*, the global evaluation of the system must be grounded primarily on the performance of *Measures*. This is because the correctness of a dimension (*Dimensions*) is inherently dependent on the correct detection and interpretation of its associated measurement (*Measures*). For instance, if three measurements are not detected, the corresponding three dimensions necessarily remain undetected, regardless of the performance of the dimension-level classifier. Similarly, cases in which a measurement is misinterpreted but yields a numerically correct value may incorrectly be treated as a correct dimension, even though the underlying measurement constitutes a false positive. Consequently, a reliable assessment of the system requires prioritizing the evaluation metrics of *Measures*, since they directly condition the downstream correctness of *Dimensions*.

Stability Analysis and Inter-trial Consistency

The robustness of the proposed architecture was evaluated through a stress-test involving 20 independent execution trials across the entire evaluation set. The resulting heatmaps reveal that model performance remains consistently high across iterations, indicating negligible variance and highly stable training behavior.

As illustrated in Figures 39 and 40—which correspond to general measure identifications and specific dimensional identifications, respectively—there are predominant yellow regions signifying a performance metric near 1.0 for the vast majority of cases.

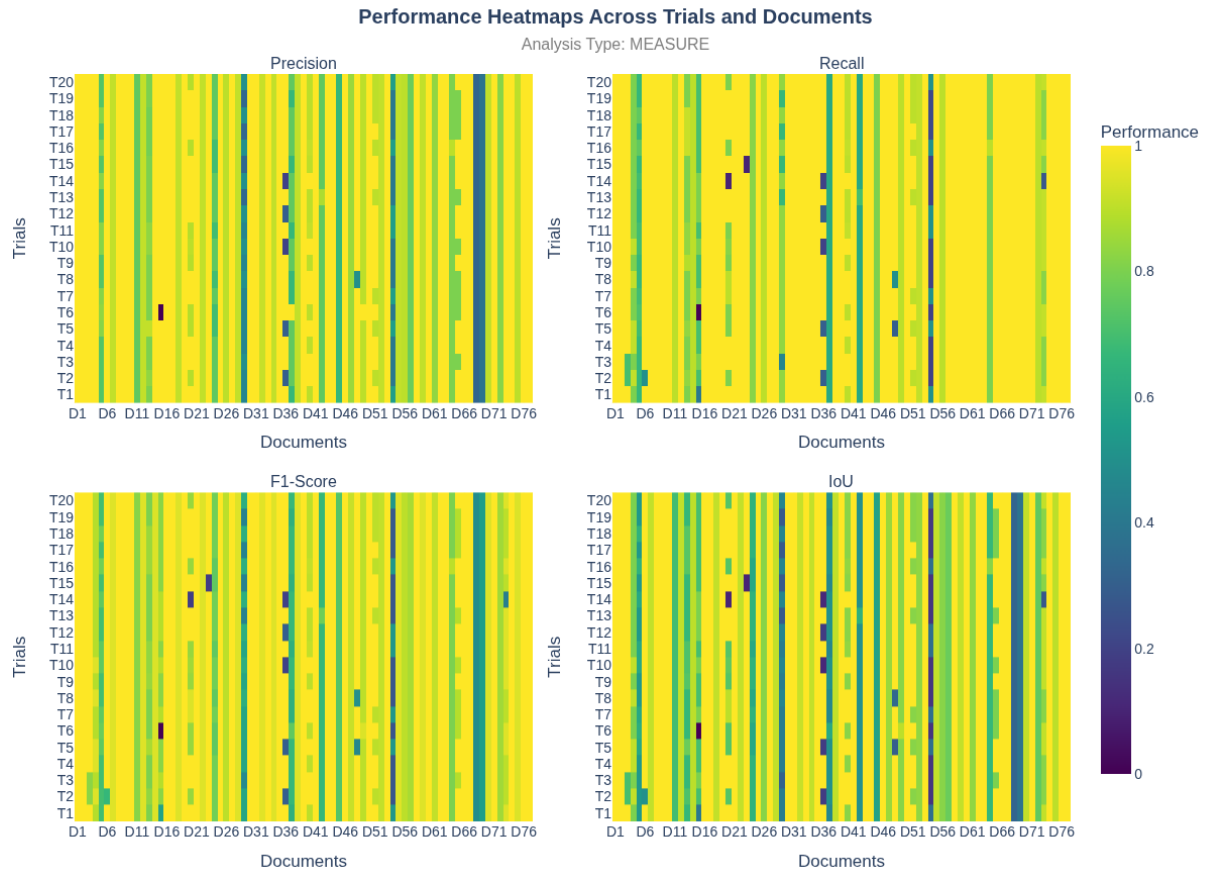


Figure 39 – Performance Heatmaps for Precision, Recall, F1-Score, and IoU across 20 trials (T1–T20) for the 77 unique documents (D1–D77). Analysis Type: MEASURE.

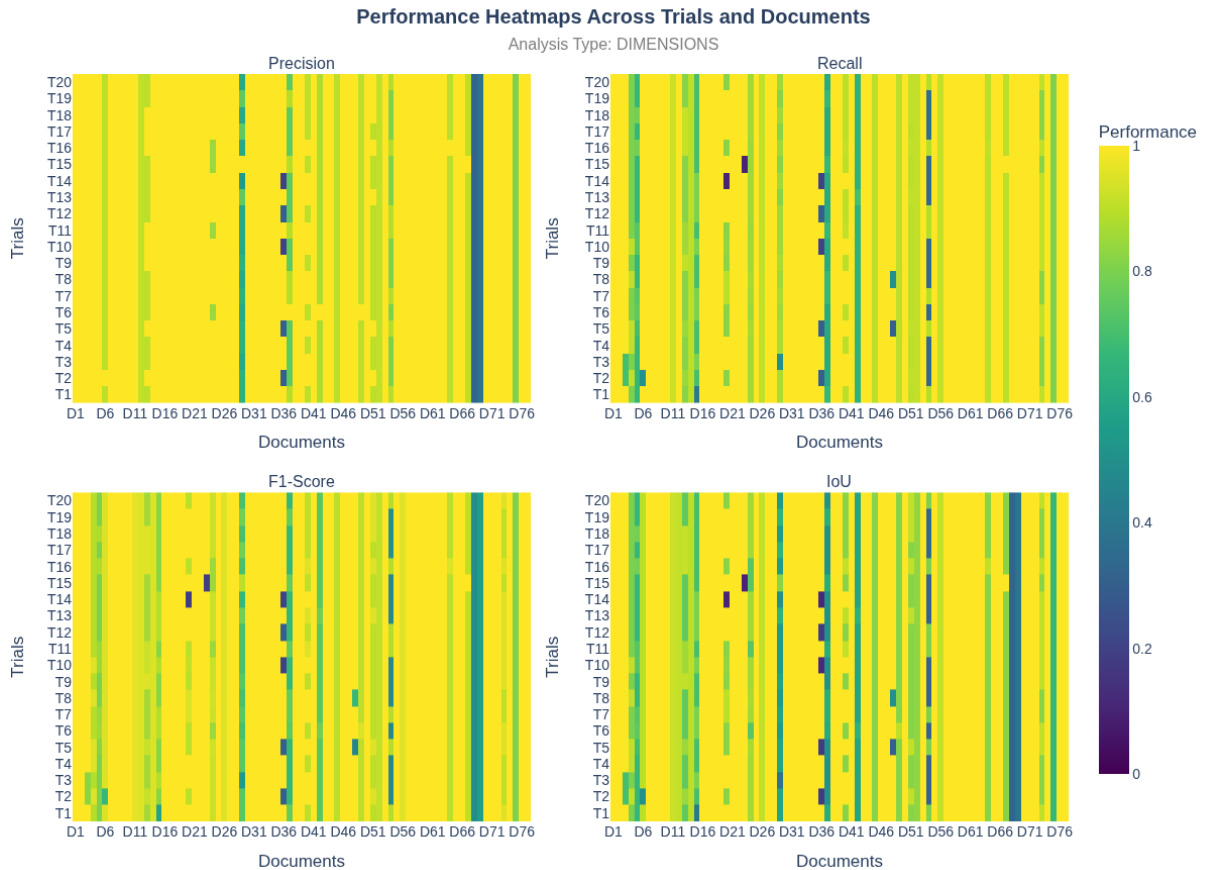


Figure 40 – Performance Heatmaps for Dimensional Extraction across 20 trials (T1–T20) for the 77 unique documents (D1–D77). Analysis Type: DIMENSIONS.

A critical observation is the presence of distinct vertical “shadows” in specific documents, such as D36 and D68. These patterns reveal that performance fluctuations are primarily **document-dependent** rather than trial-dependent. This suggests that variability is driven by intrinsic document complexities—such as high visual density, non-standard layouts, or labeling ambiguities—rather than stochastic effects during model inference.

The persistence of these vertical shadows in the dimensional analysis confirms that certain drawings contain geometric or textual layouts that are inherently challenging to parse. These specific edge cases and the nature of their visual complexities are discussed in detail in Section 5.2.

5.2 ANALYSIS OF CHALLENGES AND EDGE CASES

This section provides a comprehensive analysis of the primary challenges and edge cases encountered during the evaluation of the integrated workflow. These instances illustrate the inherent limitations and technical complexities associated with the automated extraction and verification of engineering dimensions. For the sake of

clarity and conciseness throughout this discussion, individual documents are henceforth represented using the prefix “D” followed by their corresponding index (e.g., D77).

5.2.1 Challenge: Impact of Geometric Interference on Data Extraction

Figure 41 illustrates a limitation observed in the process. Although the MLLM was able to correctly interpret the measure named *BB* (value 1326) despite the crossing line, the OCR component failed to read it under these conditions. After removing the crossing line, the OCR successfully extracted the value 1326 for *BB* measure (Figure 42). This example highlights how the process is evaluated as a whole, considering only its input and final output.

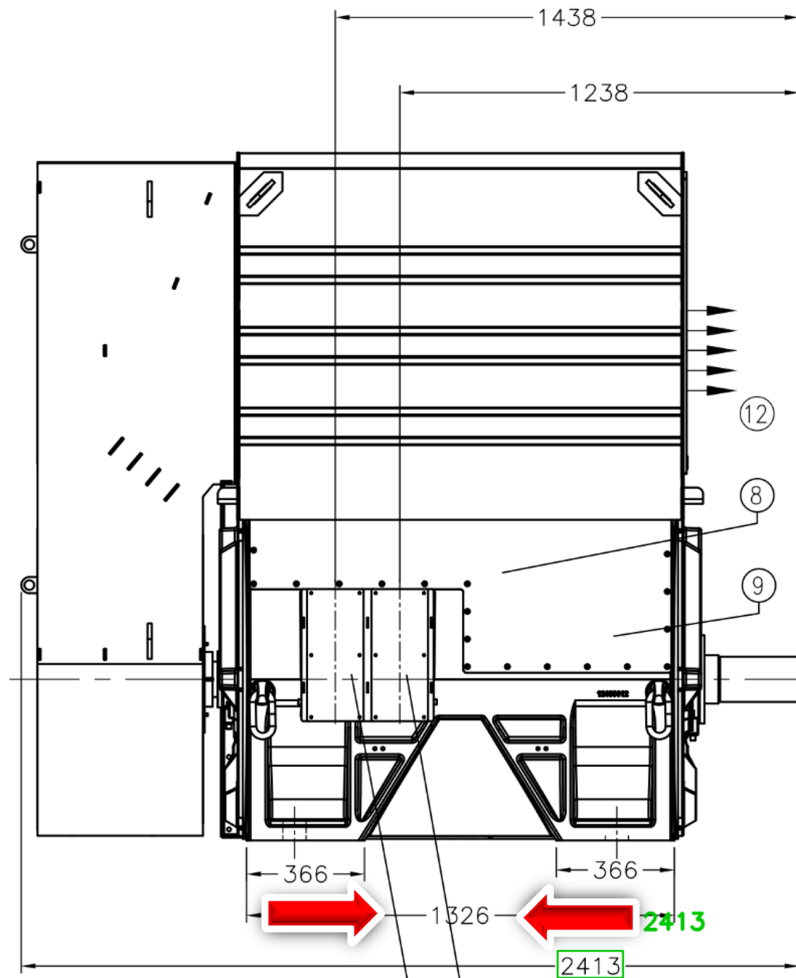


Figure 41 – Example in D39 - Pipeline failure caused by the crossing line.

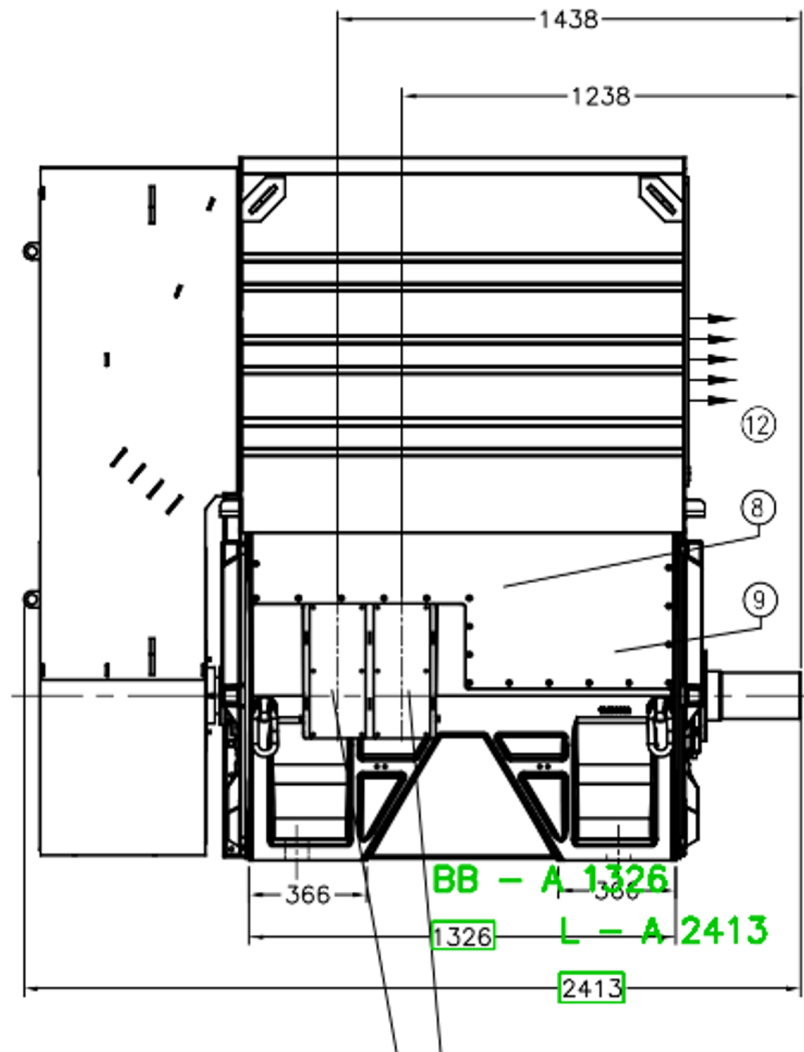


Figure 42 – Pipeline success after removing the crossing line.

Despite these obstacles, Figure 43 demonstrates that the pipeline is occasionally capable of correctly extracting information even in the presence of obstructing lines. This resilience suggests that the hierarchical structure of the four-model pipeline provides sufficient context to support the extraction of data that traditional, isolated OCR methods might fail to process.

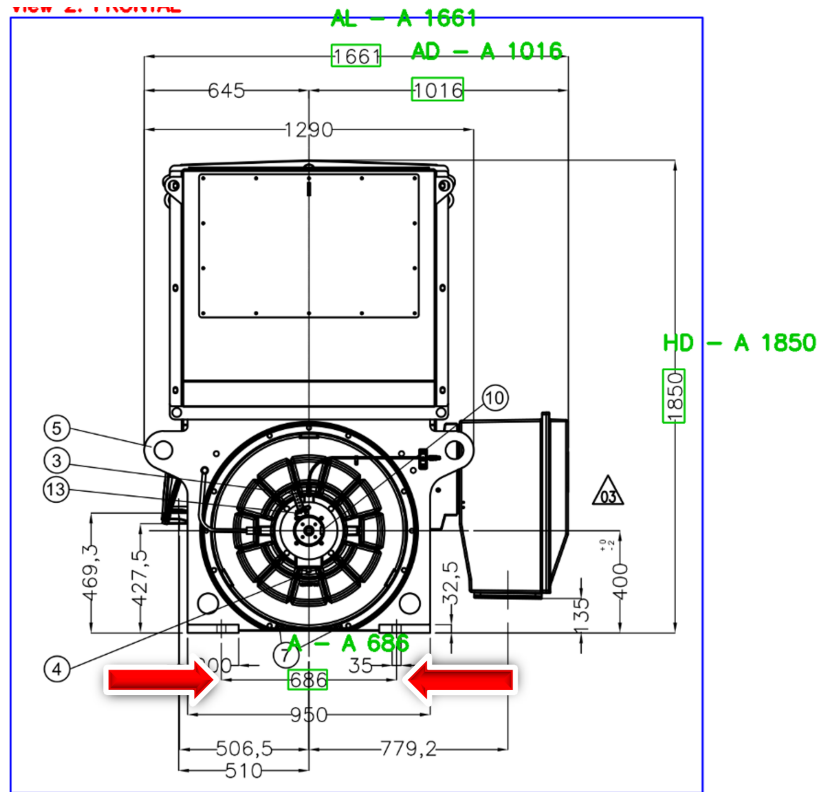


Figure 43 – Example D25 - Pipeline success without removing the crossing line.

In another successful case, the obstructing line (red arrow) did not allow for standard character recognition. However, through the use of the MLLM model, it was possible to identify the measure accurately (blue arrow). The output indicating that the pipeline successfully retrieved the information is displayed above the machine view, as shown in Figure 44.

The blue arrow in Figure 44 serves as a diagnostic overlay to confirm the extraction. It represents, in sequence: first, the **measure name** (e.g., BD); second, the **information retrieved from the database** (A); third, the **value identified by the MLLM model** (G); and finally, the **specific machine view** (V) where the information was located.

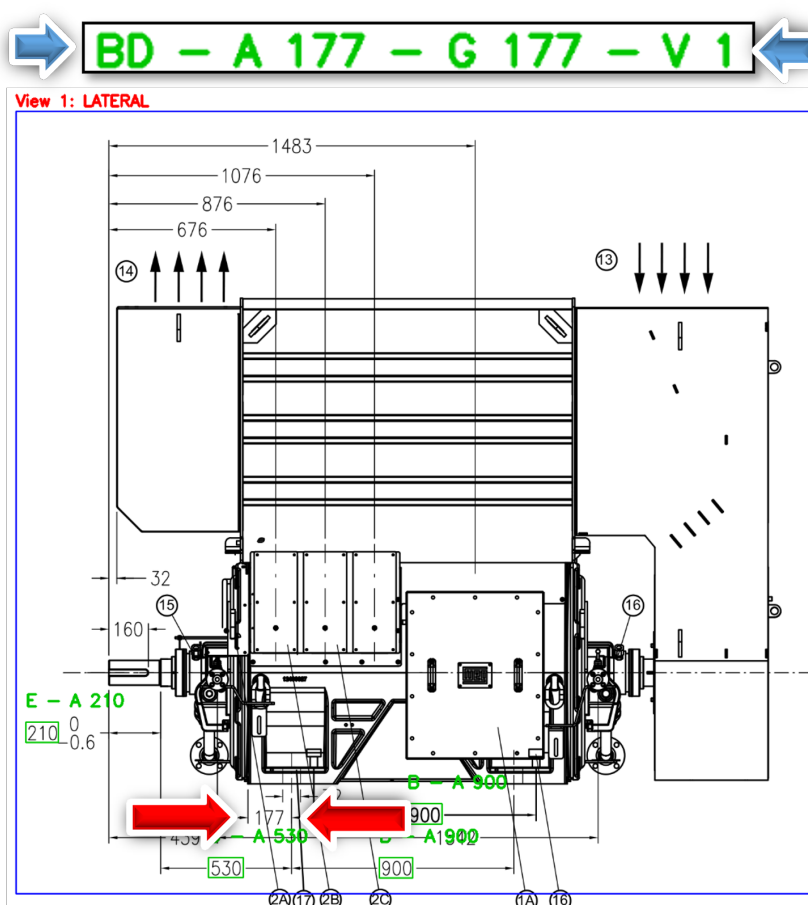


Figure 44 – Example in D32: Data retrieval achieved via MLLM reasoning despite the presence of obstructing geometry.

5.2.2 Challenge: Ambiguity in Single-Digit Character Recognition

During the analysis of D68 (Figure 45) and D69, a specific edge case was identified regarding the sensitivity of the OCR engine. While dimensions represented by multiple digits (equal to or greater than 10) or those containing decimal points were extracted with high reliability, isolated single-digit dimensions (integers from 1 to 9) presented significant interpretation challenges.

The core of the problem lies in the aspect ratio and pixel density of isolated digits. In many instances, the engine struggles to distinguish a single character from surrounding noise or geometric remnants of the drawing, leading to misinterpretations where a digit like “9” could be erroneously classified or missed entirely.

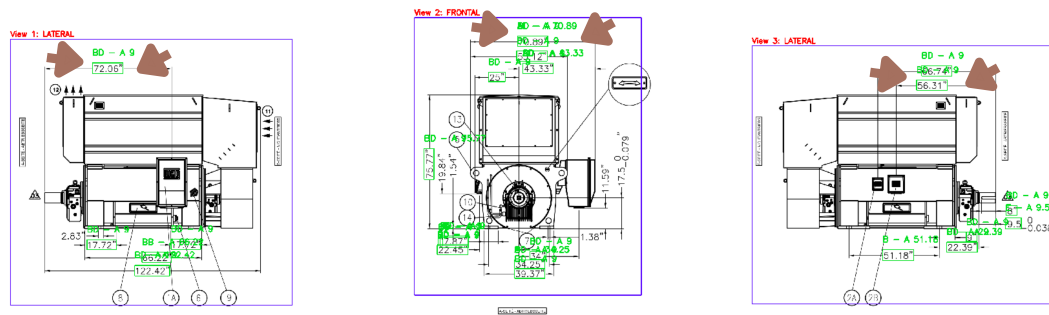


Figure 45 – Example of OCR limitation in D68: Challenges in interpreting isolated single-digit dimensions despite image preprocessing.

5.2.3 Challenge: Performance Variability and Result Oscillation

An analysis of the heatmaps in Figures 39 and 40 reveals a significant oscillation in the results for D36. While the system achieved a performance of 1.0 (or near it) in most trials, certain iterations saw a sharp decline, with scores dropping to approximately 0.4. This phenomenon is illustrated by comparing two distinct trials: Figure 46 (Trial 1), where the system reached a 1.0 score, and Figure 47 (Trial 2), where the performance fell to 0.4.

In the trials where oscillation occurred, the pipeline successfully localized the correct views for each measure; however, the semantic validation of the dimensions failed. Specifically, the system incorrectly concluded that the dimensions in the drawing matched the central database, failing to detect subtle discrepancies. Conversely, in successful trials (no oscillation), the program correctly identified these differences and visually flagged them in red.

This variability is attributed to the stochastic nature of the MLLM's interpretation layer. Even when the visual detection of the ROI (Region of Interest) is consistent, the semantic reasoning process can occasionally produce inconsistent validations, leading to false positives where a mismatch is erroneously interpreted as a correct value.

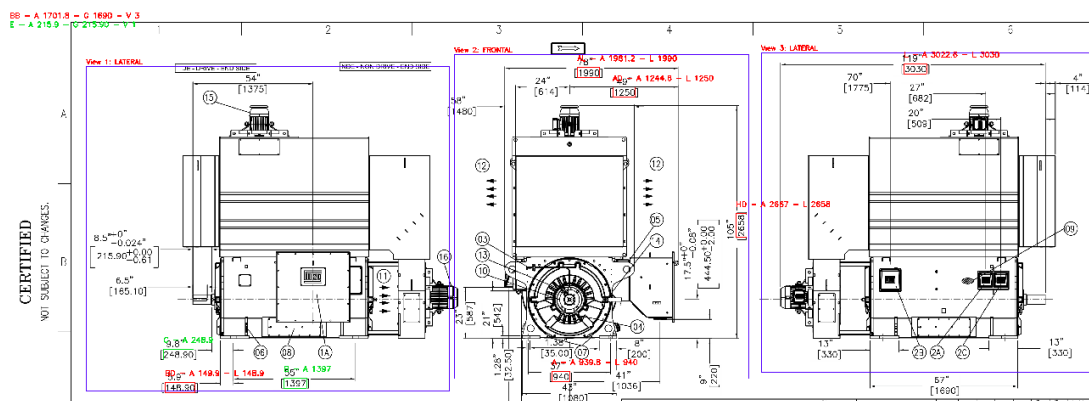


Figure 46 – Successful verification in D36 (Trial 1): The system correctly identifies and flags dimensional discrepancies.

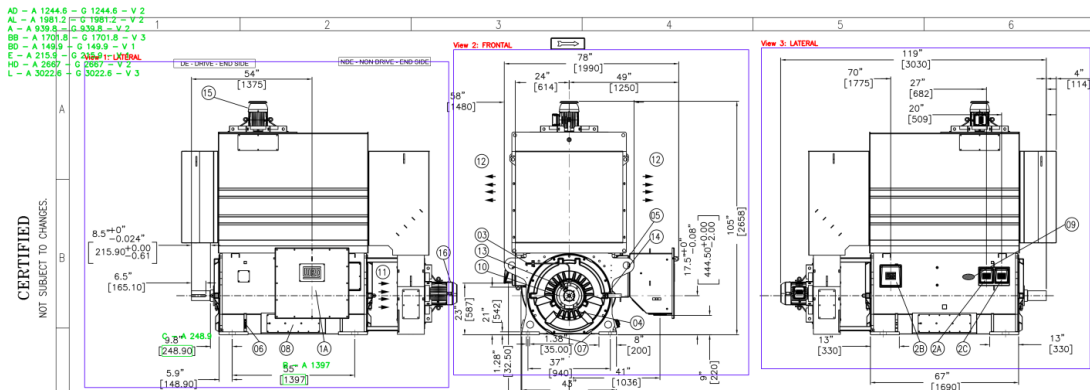


Figure 47 – Performance oscillation in D36 (Trial 2): The system fails to recognize the discrepancy despite correct view localization.

5.2.4 Challenge: Data Conflict and False Positives

Figure 48 illustrates a specific data conflict. The *AD* measure had an expected value of 849 (from the system), yet the document listed it as 1022. This discrepancy was further complicated by the presence of another, pre-existing dimension in the same view that also had the value 1022.

When the extraction process attempted to retrieve the *AD* measure, it correctly identified the value 1022, which is considered **semantically correct** since it matches the value printed on the drawing. However, the system also incorrectly matched the other pre-existing dimension (located in the lower position of the figure), treating it as a second instance of *AD*. This additional detection constitutes the false positive.

In this scenario, while the nominal measure is correct, the spatial proximity of multiple identical values can disrupt the contextual alignment, leading to a false positive mapping.

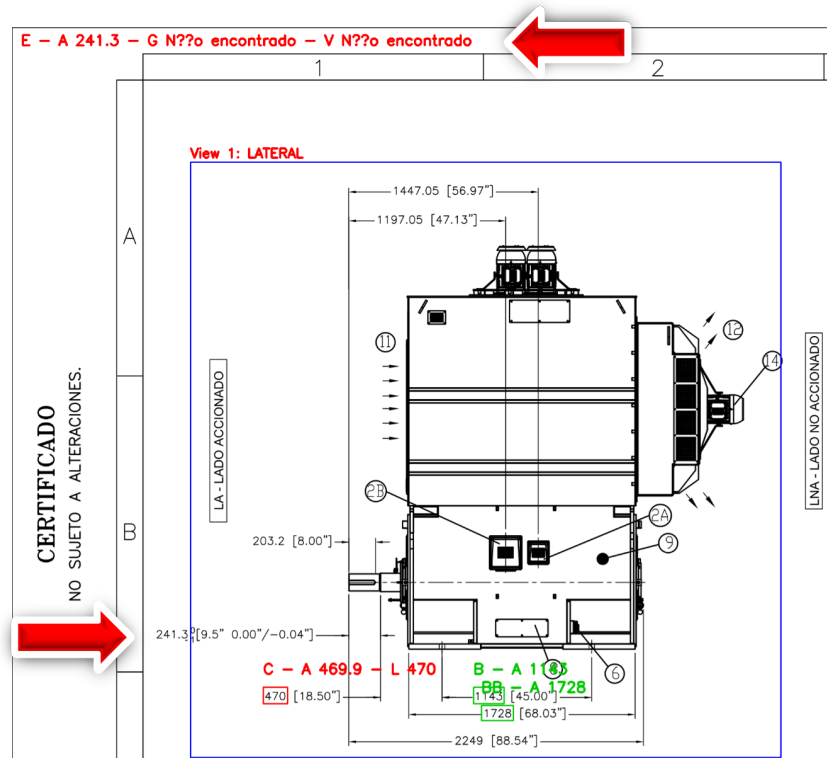


Figure 49 – Example in D56 - A detection failure.

5.2.6 Challenge: Contextual Recovery in Non-Standard Drawing Layouts

Figure 50 demonstrates that even when the initial crop from Model 1 is restrictive (indicated by the Red Arrow), the MLLM can still retrieve the required information. This is possible because the system does not send isolated views in a vacuum; instead, it sends a composite image of all views together to provide the model with global context.

However, it is important to observe that dimensions in sections like “View 2” are often located far from the primary machine body. As shown in Figure 51, while no dimensions were missed in this specific instance, the proximity of these annotations to the crop boundaries highlights a significant bottleneck. This occurs due to the lack of standardization in the placement of dimensions within industrial technical documents. Even when Model 1 identifies the machine and applies a padding adjustment to the bounding box, non-standardized layouts can still lead to the truncation of peripheral data in more complex drawings with a higher density of measures.

In this specific case, Figure 50 shows a measure that was successfully recovered in the final output (Figure 51) due to the MLLM’s reasoning (Red Arrow). Conversely, the Green Arrow highlights a situation where, even with the four-model pipeline, it was not possible to identify measure *E*. This failure likely occurred because the numerical values are positioned so closely that it led to an extraction misunderstanding. In this scenario, the MLLM proved essential by helping to locate this information and correctly

presenting it to the user in the final document.

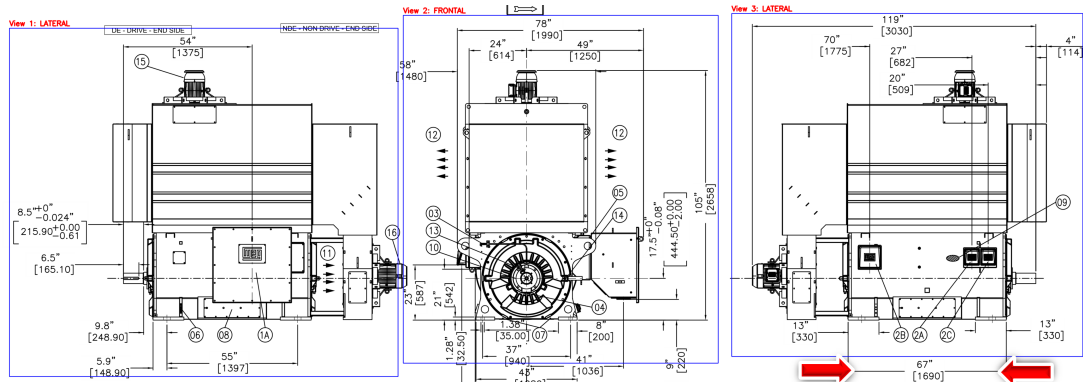


Figure 50 – Example in D36 - Composite image and context sent to the MLLM for reasoning.

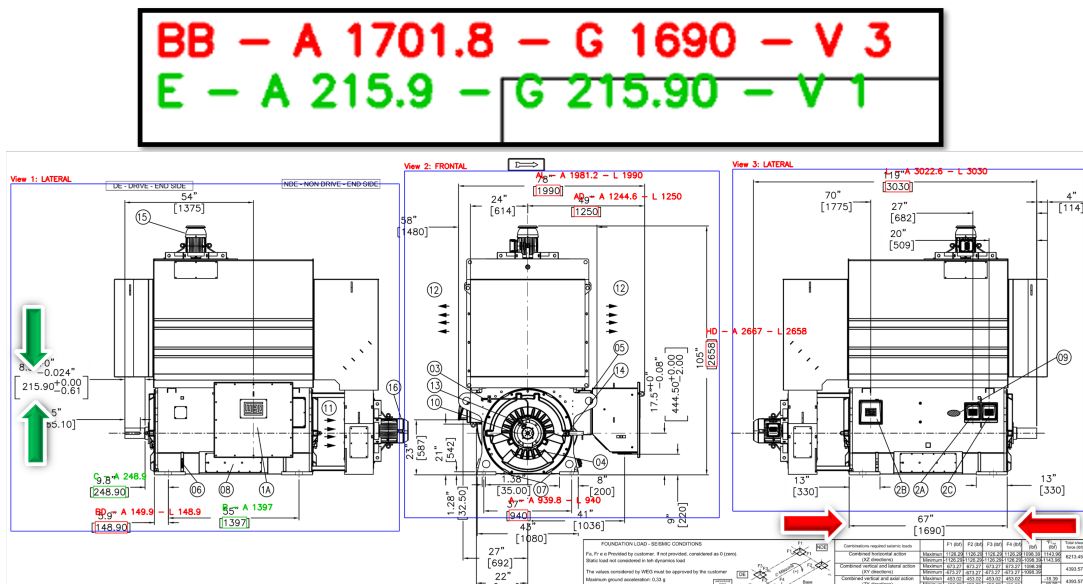


Figure 51 – Example in D36 - Final output of the pipeline. Although all measures were captured in this case, the tight ROI (Region of Interest) illustrates a potential risk for drawings with more peripheral dimensions.

5.2.7 Challenge: Semantic Misalignment and MLLM Hallucinations

Figure 52 illustrates a scenario where the MLLM's semantic reasoning failed to align correctly with the physical data present in the document. To facilitate comparison, the correct positions and names of the intended measures are indicated in the figure with green arrows. While other measures were successfully processed, this specific case focuses on five measures that highlight a semantic misunderstanding. In this instance, the final output revealed significant discrepancies: three measures were identified with critical errors, and two were missed entirely.

The three dimensions identified by the model exhibited clear misattributions:

- **Measure BD :** This was erroneously assigned to “View 4”, despite being physically located in “View 1”. Furthermore, the model reported a value that differed from the actual document content.
- **Measures AD and AL :** The MLLM incorrectly reported that the identified values (G) were identical to the database expectations (A), when in fact they differed from the values printed on the drawing. This is visible in the diagnostic information displayed beside the machine view, where the model erroneously confirms a match that does not exist in reality.

This type of error, characterized as a **semantic hallucination**, suggests that the model occasionally prioritizes internal linguistic patterns or database expectations over the literal visual evidence provided in the crop. Additionally, measures BB (located in View 3) and L (located in View 1) were not retrieved by the model at all, representing a total loss of information for these specific points.

This case underscores the distinct roles within the proposed pipeline: while the Computer Vision models are responsible for localization, the MLLM provides semantic interpretation. The information displayed above the machine views is designed to flag these discrepancies, especially when the MLLM’s findings conflict with the standard Computer Vision flow. This mechanism acts as a diagnostic tool, indicating to the user which areas require manual audit. Although these automated insights generally enhance the data extraction process, this example demonstrates that the model can occasionally misunderstand complex measures, necessitating a **human-in-the-loop** approach for final validation.

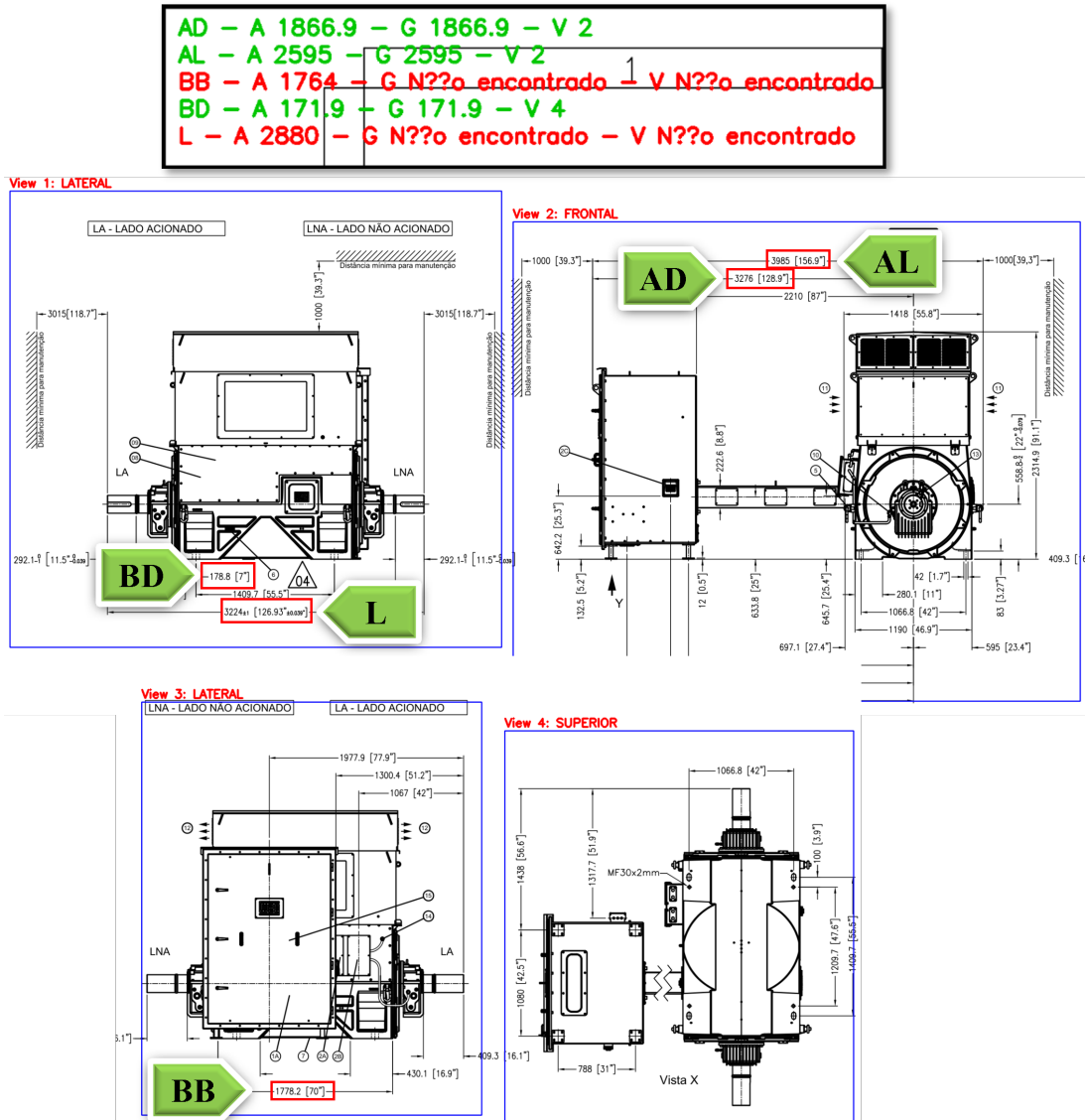


Figure 52 – Example in D5: Illustration of incorrect view attribution, value mismatch hallucinations (incorrectly flagging matches), and missed detections. Actual measures and positions are indicated by green arrows for reference.

5.3 CHAPTER CONSIDERATIONS

The results presented in this chapter demonstrate that the proposed architecture is effective in automating the verification of dimensional data in engineering drawings. The quantitative metrics confirm that the system achieves suitable precision and stability, largely driven by the robust localization capabilities of the YOLOv8 models. Furthermore, the qualitative analysis of edge cases highlights the distinct advantage of incorporating an MLLM, which provides crucial contextual recovery when traditional OCR fails due to geometric interference.

However, the evaluation also underscores persistent challenges inherent to

legacy industrial documents, such as single-digit OCR ambiguity, tight bounding box constraints, and occasional semantic hallucinations generated by the MLLM's stochastic nature. Addressing these limitations relies on the framework's human-in-the-loop diagnostic overlay, ensuring that data integrity is maintained before interacting with the Production System database. These findings establish a solid foundation for the final conclusions and future work discussed in the following chapter.

6 CONCLUSION

This research aimed to automate and enhance the verification of dimensional data in engineering drawings. It successfully developed an architecture that integrates computer vision and semantic processing to overcome the limitations of traditional OCR in dealing with visually complex technical documents.

The experimental evaluation demonstrates that the proposed approach achieves suitable reliability and operational robustness, effectively automating the dimensional verification process. Despite the inherent high entropy of engineering drawings and the distinct limitations of individual AI models, the integrated workflow delivered consistently accurate results across diverse drawing layouts.

A critical factor in this success was the implementation of the dual-verification workflow. Specifically, the strategy employs the MLLM for initial semantic identification, which is subsequently validated through a targeted OCR extraction for lexical cross-referencing. This sequential approach allowed the framework to confirm the physical presence of dimensions that the MLLM initially identified semantically, significantly reducing the rate of AI “hallucinations” and false positives.

Furthermore, a fundamental scientific finding of this study is the absolute necessity of maintaining specialized Convolutional Neural Networks (CNNs) for geometric localization, despite the rapidly advancing visual capabilities of modern MLLMs. While MLLMs demonstrate a sophisticated understanding of global context, they inherently struggle with the high-entropy and high-resolution requirements of dense technical drawings. In this architecture, the dedicated CNN-based object detection acts as a strict spatial filter, isolating critical regions of interest. This decomposition ensures that the subsequent semantic and lexical analysis is performed exclusively on high-resolution crops, preventing the degradation of data accuracy that typically occurs when processing full-scale documents globally. Ultimately, the synergy between geometric precision (CNN) and cognitive reasoning (MLLM) results in a verification system that is significantly more reliable than any standalone unimodal approach.

6.1 FUTURE WORK

Based on the findings and the specific edge cases identified during this study, several avenues for future research and continuous improvement are proposed:

- **Model 1 Refinement:** Investigate advanced data augmentation techniques or architectural adjustments to improve the mean Average Precision (mAP) of the

Global View Detector, mitigating the variance observed across complex document layouts.

- **Direct ERP Integration:** Develop a secure API interface for the automated population of Enterprise Resource Planning (ERP) systems, allowing the AI-verified dimensions to be uploaded and synchronized directly into industrial databases without requiring manual human-in-the-loop validation.

REFERENCES

- AKIBA, T. et al. Optuna: A next-generation hyperparameter optimization framework. In: **Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining**. [S.l.: s.n.], 2019. p. 2623–2631.
- BENGIO, Y. Practical recommendations for gradient-based training of deep architectures. In: **Neural Networks: Tricks of the Trade**. [S.l.]: Springer, 2012. p. 437–478.
- BRERETON, P.; KITCHENHAM, B.; BUDGEN, D. **Parsifal: A Tool for Supporting the Systematic Literature Review Process**. <<https://parsif.al>>. Acessado em junho de 2025.
- BROWNLEE, J. **Deep Learning for Natural Language Processing**. 1. ed. Vermont, Australia: Machine Learning Mastery, 2017.
- CAPES. **Tabela de Áreas de Conhecimento/Avaliação**. 2020. Publicado em 19/09/2020. Disponível em: <<https://www.gov.br/capes/pt-br/acesso-a-informacao/acoes-e-programas/avaliacao/instrumentos/documentos-de-apoio/tabela-de-areas-de-conhecimento-avaliacao>>. Acesso em: 3 jun. 2025.
- COMANICI, G. et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. **arXiv preprint arXiv:2507.06261**, 2025. Disponível em: <<https://arxiv.org/abs/2507.06261>>.
- DORIS, A. C. et al. Designqa: A multimodal benchmark for evaluating large language models' understanding of engineering documentation. **Journal of Computing and Information Science in Engineering**, ASME, 2024.
- EDEN, A. H. Three paradigms of computer science. **Minds & Machines**, v. 17, p. 135–167, 2007.
- GAO, W.; ZHAO, Y.; SMIDTS, C. Component detection in piping and instrumentation diagrams of nuclear power plants based on neural networks. **Progress in Nuclear Energy**, Elsevier, v. 128, p. 103491, 2020.
- GE, Z. et al. YoloX: Exceeding yolo series in 2021. In: **arXiv preprint arXiv:2107.08430**. [S.l.: s.n.], 2021.
- GIESECKE, F. E. et al. **Technical Drawing with Engineering Graphics**. Fourteenth. Harlow, Essex: Pearson Education Limited, 2014. ISBN 978-1-292-02618-3.
- GIL, A. C. **Como Elaborar Projetos de Pesquisa**. 5. ed. São Paulo: Atlas, 2010. ISBN 978-85-224-5823-3.
- GONZALEZ, R. C.; WOODS, R. E. **Digital Image Processing**. 4th. ed. [S.l.]: Pearson, 2018.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. **The Elements of Statistical Learning: Data Mining, Inference, and Prediction**. 2nd. ed. [S.l.]: Springer Science & Business Media, 2009.

HERFANDI, H. et al. Real-time patient indoor health monitoring and location tracking with optical camera communications on the internet of medical things. **Sensors**, MDPI, v. 23, n. 18, p. 7905, 2023.

HU, H.; ZHANG, C.; LIANG, Y. Detection of surface roughness of mechanical drawings with deep learning. **Journal of Mechanical Science and Technology**, Springer, v. 35, p. 5541–5549, 2021.

HUTTER, F.; HOOS, H.; LEYTON-BROWN, K. An efficient approach for assessing hyperparameter importance. In: PMLR. **International Conference on Machine Learning (ICML)**. [S.l.], 2014. p. 754–762.

JANG, S.; ROH, H.; LEE, G. Generative ai in architectural design: Application, data, and evaluation methods. **Automation in Construction**, Elsevier, v. 174, p. 106174, 2025. Disponível em: <<https://doi.org/10.1016/j.autcon.2025.106174>>.

JOCHER, G.; CHAURASIA, A.; JING, Q. **Ultralytics YOLOv8**. GitHub, 2023. Disponível em: <<https://github.com/ultralytics/ultralytics>>.

JOCHER, G. et al. **Ultralytics YOLOv5**. Zenodo, 2020. Disponível em: <<https://github.com/ultralytics/yolov5>>.

JR., W. P.; PEREIRA, V. L. D. d. V.; FILHO, H. d. V. P. **Pesquisa Científica sem Tropeços: Abordagem Sistêmica**. São Paulo: Atlas, 2007. ISBN 978-85-224-4843-2.

KAGERMANN, H. et al. **Recommendations for implementing the strategic initiative INDUSTRIE 4.0: Final report of the Industrie 4.0 Working Group**. [S.l.], 2013.

KHALLOULI, W. et al. Leveraging transformer-based ocr model with generative data augmentation for engineering document recognition. **Electronics**, MDPI, v. 14, n. 1, p. 5, 2024.

KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. **arXiv preprint arXiv:1412.6980**, 2014.

KITCHENHAM, B. et al. Guidelines for performing systematic literature reviews in software engineering. **EBSE Technical Report**, Keele University and Durham University Joint Report, v. 2, n. 1, p. 1–57, 2007.

KOHAVI, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: **Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)**. [S.l.: s.n.], 1995. v. 2, p. 1137–1143.

LI, X. et al. Generalized focal loss: Learning qualified predictive density for object detection. In: **Advances in Neural Information Processing Systems (NeurIPS)**. [S.l.: s.n.], 2020. v. 33, p. 21002–21012.

- LIANG, X.; WANG, Z.; LIU, J. Towards a self-cognitive complex product design system: A fine-grained multi-modal feature recognition and semantic understanding approach using large language models in mechanical engineering. **Advanced Engineering Informatics**, Elsevier, v. 65, p. 103265, 2025.
- LIN, T.-Y. et al. Focal loss for dense object detection. In: . [S.l.: s.n.], 2017. p. 2980–2988. Foundational theory for Classification Loss in dense detectors.
- LIN, T.-Y. et al. Microsoft coco: Common objects in context. In: SPRINGER. **European Conference on Computer Vision (ECCV)**. [S.l.], 2014. p. 740–755.
- LIN, Y.-H. et al. Integration of deep learning for automatic recognition of 2d engineering drawings. **Machines**, v. 11, n. 8, p. 802, 2023.
- LIU, S. et al. Path aggregation network for instance segmentation. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2018. p. 8759–8768.
- LOSHCHILOV, I.; HUTTER, F. Decoupled weight decay regularization. In: **International Conference on Learning Representations (ICLR)**. [S.l.: s.n.], 2019.
- LUNDBERG, S. M.; LEE, S.-I. A unified approach to interpreting model predictions. In: **Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)**. [S.l.: s.n.], 2017. p. 4768–4777.
- MARCONI; LAKATOS. **Fundamentos da Metodologia Científica**. 6. ed. [S.l.]: Editora Atlas, 2005.
- MAUPOU, C. et al. Automatic raster engineering drawing digitisation for legacy parts towards advanced manufacturing. **Procedia CIRP**, Elsevier, v. 129, p. 234–239, 2024.
- MORENO-GARCÍA, C. F.; ELYAN, E.; JAYNE, C. New trends on digitisation of complex engineering drawings. **Neural Computing and Applications**, Springer, v. 31, n. 5, p. 1695–1712, 2019. Disponível em: <<https://doi.org/10.1007/s00521-018-3583-1>>.
- MORI, S.; SUEN, C. Y.; YAMAMOTO, K. Historical review of ocr research and development. **Proceedings of the IEEE**, v. 80, n. 7, p. 1029–1058, 1992.
- NAGARHALLI, T. P.; VAZE, V.; RANA, N. K. Impact of machine learning in natural language processing: A review. In: IEEE. **2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)**. Tirunelveli, India, 2021. p. 1529–1534.
- NGO, A. Q.; NGUYEN, L. Q.; TRAN, V. Q. Developing interpretable machine learning-shapley additive explanations model for unconfined compressive strength of cohesive soils stabilized with geopolymer. **PLOS ONE**, Public Library of Science, v. 18, n. 6, p. e0286950, 2023. Disponível em: <<https://doi.org/10.1371/journal.pone.0286950>>.
- O'SHEA, K.; NASH, R. An introduction to convolutional neural networks. **arXiv preprint arXiv:1511.08458**, 2015. Disponível em: <<https://arxiv.org/abs/1511.08458>>.
- POWERS, D. M. W. Evaluation: From precision, recall and f-measure to roc, informedness, markedness and correlation. **Journal of Machine Learning Technologies**, v. 2, n. 1, p. 37–63, 2011.

PRECHELT, L. Early stopping-but when? In: **Neural Networks: Tricks of the trade**. [S.l.]: Springer, 2002. p. 55–69.

PRESS, G. Cleaning big data: Most time-consuming, least enjoyable data science task, survey says. **Forbes Magazine**, 2016.

PRODANOV, C. C.; FREITAS, E. C. de. **Metodologia do Trabalho Científico: Métodos e Técnicas de Pesquisa e do Trabalho Acadêmico**. 2. ed. Novo Hamburgo: Feevale, 2013. 276 p. ISBN 978-85-7717-158-3.

RASCHKA, S. Model evaluation, model selection, and algorithm selection in machine learning. **arXiv preprint arXiv:1811.12808**, 2018.

REASON, J. **Human Error**. Cambridge: Cambridge University Press, 1990.

REDMON, J. et al. You only look once: Unified, real-time object detection. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2016. p. 779–788.

REN, S. et al. Faster r-cnn: Towards real-time object detection with region proposal networks. In: **Advances in Neural Information Processing Systems (NeurIPS)**. [S.l.: s.n.], 2015.

REZATOFIGHI, H. et al. Generalized intersection over union: A metric and a loss for bounding box regression. In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2019. p. 658–666.

ROBBINS, H.; MONRO, S. A stochastic approximation method. **The Annals of Mathematical Statistics**, JSTOR, v. 22, n. 3, p. 400–407, 1951.

SAMBASIVAN, N. et al. "everyone wants to do the model work, not the data work": Data centric ai in critical domains. In: **proceedings of the 2021 CHI Conference on Human Factors in Computing Systems**. [S.l.: s.n.], 2021. p. 1–15.

SCHEIBEL, B.; MANGLER, J.; RINDERLE-MA, S. Extraction of dimension requirements from engineering drawings for supporting quality control in production processes. **Computers in Industry**, Elsevier, v. 129, p. 103442, 2021.

SCHLAGENHAUF, T.; NETZER, M.; HILLINGER, J. Text detection on technical drawings for the digitization of brown-field processes. **Procedia CIRP**, Elsevier, v. 118, p. 372–377, 2023.

SCHÖNFELDER, P. et al. Deep learning-based text detection and recognition on architectural floor plans. **Automation in Construction**, Elsevier, v. 157, p. 105156, 2024.

SMITH, R. An overview of the tesseract OCR engine. In: **Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR)**. [S.l.: s.n.], 2007. p. 629–633.

STONE, M. Cross-validatory choice and assessment of statistical predictions. **Journal of the Royal Statistical Society: Series B (Methodological)**, v. 36, n. 2, p. 111–133, 1974.

TERVEN, J.; CÓRDOVA-ESPARZA, D.-M. A comprehensive review of yolo: From yolov1 to yolov8 and beyond. **arXiv preprint arXiv:2304.00501**, 2023.

TORO, J. V.; TARKIAN, M. Optimizing text recognition in mechanical drawings: A comprehensive approach. **Available at SSRN 5045921**, 2025.

WANG, X.; ZHOU, Z.-H.; OTHERS. Analysis and recognition of 2d engineering drawings: A review. **Computer-Aided Design**, Elsevier, v. 130, 2020.

WANG, Z. et al. Towards cognitive intelligence-enabled product design: The evolution, state-of-the-art, and future of ai-enabled product design. **Journal of Industrial Information Integration**, Elsevier, v. 43, p. 100759, 2025. Disponível em: <<https://doi.org/10.1016/j.jii.2024.100759>>.

WAZLAWICK, R. S. Uma reflexão sobre a pesquisa em ciência da computação à luz da classificação das ciências e do método científico. **Revista de Sistemas de Informação da FSMA**, n. 6, p. 3–10, 2010.

XU, Y. et al. Tolerance information extraction for mechanical engineering drawings—a digital image processing and deep learning-based model. **CIRP Journal of Manufacturing Science and Technology**, Elsevier, v. 50, p. 55–64, 2024.

XUESONG, X. et al. Associative reasoning for engineering drawings using an interactive attention mechanism. **Automation in Construction**, Elsevier, v. 170, p. 105942, 2025.

ZHANG, W. et al. Component segmentation of engineering drawings using graph convolutional networks. **Computers in Industry**, Elsevier, v. 147, p. 103885, 2023.

ZHENG, Z. et al. Distance-iou loss: Faster and better learning for bounding box regression. In: **Proceedings of the AAAI Conference on Artificial Intelligence**. [S.l.: s.n.], 2020. v. 34, n. 07, p. 12993–13000.

ZHONG, R. Y. et al. Intelligent manufacturing in the context of industry 4.0: A review. **Engineering**, Elsevier, v. 3, n. 5, p. 616–630, 2017.

ZOU, Z. et al. Object detection in 20 years: A survey. **Proceedings of the IEEE**, IEEE, v. 111, n. 3, p. 257–278, 2023.

A APPENDIX

A.1 MLLM PROMPT GENERATION FUNCTION

This Python function generates a detailed prompt for a Multimodal Large Language Model (MLLM). The model acts as a design engineer responsible for interpreting and extracting specific dimensions from technical drawings of electrical machines.

```

1 def prompt_LLM(
2     view_list_for_prompt,
3     construction_form,
4     value_list
5 ):
6
7     prompt_text = f'''
8
9         Context:
10        You are a design engineer with extensive experience in
11        electrical machines, specializing in reading
12        and interpreting technical drawings.
13
14        Description:
15        In this context,
16        **{len(view_list_for_prompt)} views of electrical machines**
17        have been identified in the drawing.
18        Some of them may have specific tolerances.
19        Each view is named as follows:
20            - View No. - View Type (FRONT, SIDE, TOP)
21
22        Objective:
23        Interpret each dimension related to the **machine's envelope**,
24        based on its technical semantics.
25
26        **Machine construction form:** {construction_form}
27            - The main terminal box is on the {construction_form[-1]}.
28            > Note: 'L' means **Left** and 'R' means **Right**.
29
30        System Input Values:
31        {value_list}
32
33        Data Format:
34            - First piece of information: Variable name.
35            - Second piece of information: Expected value (ASSOM).
36            - The values in the document may be correct or divergent.
37            - Verification must be performed

```

```

38     **first based on the variable's semantics**,
39     and then on the coherence of the identified value.
40
41     Task:
42     For each variable present in the document, return it
43     in the following format:
44
45     Response Format:
46     Variable: VARIABLE_NAME
47     ASSOM: 'EXPECTED_VALUE'
48     GenAI: 'FOUND_VALUE'
49     View: 'View Number'
50     Observation: [if necessary      view, tolerance, ambiguity, etc.]
51
52     Example:
53     Variable: ZMEDIDA_H_ASSOM_WM_03
54     ASSOM: '27'
55     GenAI: '24'
56     View: '1'
57     Observation: This dimension has a tolerance of 0"/-0.078"
58     indicated on the drawing.
59
60     **Additional rules for divergence detection:**
61     - Before comparing values, remove symbols like quotes ('"')
62     and extra spaces.
63     - Consider 'ASSOM' and 'GenAI' as correct if they
64     are **numerically equal**,
65     even if:
66         - They have a different number of
67         decimal places ('9.5' vs '9.50').
68         - They may or may not have the inch symbol
69         ('"') at the end.
70     - If there is no match by **semantics** or by
71     the **system's numerical value**,
72     return: **Not found**.
73
74     Final Notes:
75     - Respond **only for the variables found** in the document.
76     - Maintain the **exact format** requested.
77     - Use your technical judgment to validate or correct what
78     the AI has interpreted.
79     '''
80
81     regex_pattern = r"""
82     Variable:\s*(?P<variable>\S+)\s*
83     ASSOM:\s*'?(?P<assom>[\d\.]+)'?\s*
84     GenAI:\s*'?(?P<genai>[\d\.]+)'?\s*

```



```

85 View:\s*'?(?P<view>\d+)'?\s*
86 (?:Observation:\s*(?P<obs>.*?))?
87 (?:\nVariable:|\Z)
88 ""
89
90 return prompt_text, regex_pattern

```

A.2 ANALYZED MEASURES AND DEFINITIONS

The following list presents standardized variable names and their corresponding technical definitions, serving as a reference for interpreting the machine's dimensional parameters.

ZMEDIDA_BB_ASSOM_WM_03 Usable length of the machine frame base, measured between the inner support surfaces (excluding extensions and external supports). (*SIDE or TOP View*)

ZMEDIDA_L_ASSOM_WM_03 Overall length of the machine. (*SIDE View*)

ZMEDIDA_HD_ASSOM_WM_03 Overall height, from the base to the highest point. (*FRONT View*)

ZMEDIDA_A_ASSOM_WM_03 Distance between mounting holes on the base. (*FRONT or TOP View*)

ZMEDIDA_AD_ASSOM_WM_03 Distance from the shaft centerline to the **top** end of the **main terminal box**. (*FRONT View*)

- L (Left): measurement to the left side of the main terminal box.
- R (Right): measurement to the right side of the main terminal box.
- Relation: $AD = AL - AD1$

ZMEDIDA_AD1_ASSOM_WM_03 Distance from the shaft centerline to the **top** end of the **secondary terminal box**. (*FRONT View*)

- L (Left): measurement to the right side of the secondary terminal box.
- R (Right): measurement to the left side of the secondary terminal box.
- Relation: $AD1 = AL - AD$

ZMEDIDA_AL_ASSOM_WM_03 Total width of the **upper part of the machine**, given by $AL = AD + AD1$. (*FRONT View*)

ZMEDIDA_E_ASSOM_WM_03 Shaft extension length. (*SIDE or TOP View*)

ZMEDIDA_LC_ASSOM_WM_03 Overall length of the machine **with two shaft extensions**. (*SIDE View*)

ZMEDIDA_B_ASSOM_WM_03 Distance between the centers of the front and rear mounting feet holes. (*SIDE or TOP View*)

ZMEDIDA_BD_ASSOM_WM_03 Distance from the side of the frame to the center of the front foot hole. (*SIDE or TOP View*)

ZMEDIDA_C_ASSOM_WM_03 Distance between the end of the shaft extension (drive end) and the center of the front feet holes. (*SIDE or TOP View*)