big data and cognitive computing

MDPI

Article

Leveraging Large Language Models and Object Detection for Automated Knowledge Graph Generation from Industrial Schematics

Federico Lopomo [1], Valentina Faraco [1], Davide Marche [1], Saverio Ieva [2,3], Giuseppe Loseto [3,4,*], Davide Loconte [2], Floriano Scioscia [2,3] and Michele Ruta [2,3]

1 Lutech S.p.A., Via Orfeo Mazzitelli 258/E, I-70124 Bari, Italy; federico.lopomo@lutech.it (F.L.); valentina.faraco@lutech.it (V.F.); davide.marche@lutech.it (D.M.)
2 Department of Electrical and Information Engineering, Polytechnic University of Bari, Via E. Orabona 4, I-70125 Bari, Italy; saverio.ieva@poliba.it (S.I.); davide.loconte@poliba.it (D.L.); floriano.scioscia@poliba.it (F.S.); michele.ruta@poliba.it (M.R.)
3 donkeyPower S.r.l., Via E. Orabona 4, I-70125 Bari, Italy
4 Department of Engineering, LUM "Giuseppe Degennaro" University, S.S. 100 Km 18, I-70010 Casamassima, Italy
* Correspondence: loseto@lum.it; Tel.: +39-080-5963515

Abstract

Industrial digitalization increasingly requires automated tools capable of extracting structured knowledge from complex engineering documentation, such as Piping and Instrumentation Diagrams (P&IDs). This work proposes an integrated framework that combines object detection and Large Language Models (LLMs) for automated Knowledge Graph (KG) generation. The approach enables the transformation of unstructured P&ID schematics into machine-interpretable representations, supporting data-driven analysis and decision-making. A modular pipeline is developed, including image pre-processing, symbol detection via a YOLO-based model, and identification of semantic relations between schematic elements using LLMs. The proposal also includes the definition of a reference ontology, which is exploited for the construction of the KG, and a diagram dataset designed to test the performance of the object detection model. The KG generation procedure achieves strong results in terms of image reconstruction across a wide set of industrial schematics, while also preserving the semantic integrity and completeness of the original diagrams. The proposed method represents a significant step toward the digitalization of industrial knowledge, bridging traditional engineering documentation and semantic-based technologies.

Keywords: large language models; knowledge graph; object detection; piping and instrumentation diagrams; industrial digitalization; semantic reasoning; digital twin

Academic Editors: Amna Dridi, Edlira Kalemi Vakaj and Mohamed Medhat Gaber

Received: 24 March 2026
Revised: 3 June 2026
Accepted: 18 June 2026
Published: 29 June 2026

1. Introduction and Motivation

Digital transformation is reshaping industrial scenarios by driving the adoption of innovative digitalization procedures and techniques that enhance operational efficiency, data-driven decision-making, and system integration [1]. In particular, industrial digitalization refers to the integration of advanced digital technologies, such as Artificial Intelligence (AI), cloud computing, and data analytics, into industrial operations [2,3]. By leveraging digital tools, companies can streamline operations, minimize downtime, and reduce operational costs while improving overall productivity. A key component of industrial digitalization is the ability to extract, analyze, and use data usually defined in complex industrial documentation, such as Piping and Instrumentation Diagrams (P&IDs). P&IDs are essential technical drawings used in industrial processes to visually represent the relationships between piping, equipment, and instrumentation components within a system [4]. These diagrams serve as a reference model for designing and maintaining industrial plants, providing engineers and operators with critical insights into system functionality. Each symbol and connection in a P&IDs conveys specific information, such as pipe sizes, flow directions, pressure ratings, control valves, and measurement instruments. Traditionally, analyzing these diagrams has relied heavily on manual interpretation, making the process time-consuming, error-prone, and difficult to scale [5]. Object detection, powered by advanced Machine Learning (ML) techniques, represents a transformative solution to automate the extraction of meaningful entities and relationships from P&IDs schematics. Leveraging computer vision algorithms, object detection systems can identify symbols, text annotations, and structural patterns within these diagrams, converting static visual information into structured digital datasets. Basically, these systems utilize deep learning architectures [6], such as Convolutional Neural Networks (CNNs) and transformer-based models, to achieve high accuracy in recognizing diverse elements, even in low-resolution or noisy diagrams. Additionally, object detection models can be trained on domain-specific datasets to improve their performance in recognizing industry-standard symbols and complex arrangements unique to P&IDs.

However, the true value of this extracted data lies in its integration into a Knowledge Graph (KG), a powerful representation framework that organizes information into interconnected entities and relationships. KGs go beyond traditional relational databases by representing data in a graph structure, where nodes represent entities (e.g., equipment, pipelines, sensors) and edges represent the relationships between them (e.g., connection, control, measurement) [7]. This structure allows for a more flexible, semantic, and scalable representation of complex systems, resulting in being particularly valuable in environments with diverse systems and standards. KGs can unify heterogeneous data sources, bridging gaps between isolated datasets and providing a single source of truth for industrial operations [8]. KGs also enable advanced querying and inference. Users can perform complex queries across interconnected datasets, revealing hidden patterns, dependencies, and anomalies that might not be apparent in tabular data [9]. Another significant advantage is their capability to support reasoning over data, enabling systems to infer new knowledge based on existing relationships and rules. In addition, a KG is highly interpretable and human-readable. Their intuitive representation makes it easier for engineers, operators, and decision-makers to understand and interact with complex datasets, fostering better collaboration and decision-making [10]. Nevertheless, manually constructing KGs is a resource-intensive process, especially in industrial settings where datasets are vast, complex, and frequently updated. Some key challenges include: (a) integration of data coming from several sources and represented in different formats into a unified structure; (b) presence of inconsistencies and unreliable insights due to human errors during the data extraction and KG modeling tasks; (c) continuous effort required for keeping KGs up to date with evolving data and documentation.

These challenges make manual construction infeasible and highlight the need for automated approaches aiming to reduce the manual effort required to translate unstructured data into a structured format [11]. As highlighted in [12], automated KG generation also ensures scalability, allowing large repositories of existing information to be processed efficiently. Additionally, automated workflows minimize human error, improving the reliability and accuracy of the KG. These pipelines can exploit rule-based systems, machine learning models, and Natural Language Processing (NLP) techniques to ensure the accuracy and consistency of the generated data [13]. In particular, the automation of KG generation from data extracted via object detection in industrial schematics represents a significant advancement in digitalization workflows. In this process, the raw data extracted from engineering drawings, such as symbols, labels, and annotations, are transformed into structured representations suitable for KG construction. Object detection outputs, including identified entities and their relationships, are processed through entity recognition and relationship extraction pipelines. Entities such as valves, sensors, pumps, and pipelines are mapped to predefined ontologies, standardizing the representation across different diagrams.

In the context of industrial documentation, Large Language Models (LLMs) represent a novel technology that can be used to manage both structured data stored in a KG and unstructured textual sources (e.g., technical manuals, annotations, and operational guidelines) [14,15]. They can perform tasks such as semantic text analysis, summarization, and disambiguation providing contextual clarification for symbols or annotations that may have multiple interpretations based on their placement or associated metadata. Additionally, LLMs can enrich KGs by extracting new relationships and contextual information from documentation and historical records [16,17]. This semantic enrichment enhances the overall accuracy and usability of the KG also enabling a deeper understanding of system interactions, dependencies, and potential failure points. Following this vision, this paper proposes a novel pipeline combining:

- object detection for P&IDs analysis;
- automated Knowledge Graph generation;
- semantic enhancement through LLMs.

The proposed approach aims to streamline industrial documentation processing, reduce human intervention, and unlock new possibilities for data-driven decision-making in industrial environments. This automated approach empowers industries to gain valuable insights, optimize maintenance schedules, predict equipment failures, and enhance decision-making processes by leveraging the interconnected knowledge stored within the graph.

The remainder of this paper is organized as follows. Section 2 reviews recent advances about object detection applied to industrial schematics and summarizes current research efforts on integrating LLMs with KGs. Section 3 presents the proposed reference framework, detailing both the overall architecture and the underlying ontology modeling. Section 4 discusses the experimental results and examines the main challenges and limitations encountered in applying the proposed approach. Finally, Section 5 concludes the paper and outlines directions for future work.

2. Related Work

This section examines recent methods for visual interpretation of industrial schematics (e.g., P&IDs, electrical single-line diagrams) and advances that integrate LLMs with KGs to assist in industrial decision-making. The review covers three primary methodological areas: (i) detection of symbols and lines; (ii) comprehension of diagrams with reconstruction of topological elements; and (iii) LLM–KG methodologies for the extraction, retrieval, and reasoning of knowledge. The discussion for each topic addresses problem settings, model assumptions, datasets, and evaluation protocols, emphasizing both advantages and challenges in relation to drafting variability, small-object recall, text–graphics association, scalability for extensive archives, and reproducibility.

2.1. Object Detection Techniques for Industrial Schematics

Object detection in industrial schematics aims to convert raster drawings into structured, machine-usable representations while coping with dense symbol layouts, thin and

https://doi.org/10.3390/bdcc10070210

partially occluded lines, heterogeneous drafting styles, and mixed text–graphics regions. Within this context, a first group of contributions focuses specifically on P&IDs. A detector leveraging deep learning and custom image-processing rules is described in [18], designed to identify linear elements such as process lines, connectors, and flow arrows. This method achieves high precision and recall for line categories, though it only partially maps to device-level semantics. An image-based P&ID pipeline presented in [19] enables the extraction of symbols, tags, and lines without training data by combining Optical Character Recognition (OCR) and traditional vision techniques, resulting in fast end-to-end runtimes on real plant documents. Consequently, the framework introduced in [20] enhances continuous-line detection reliability, even with short linear segments and oblique strokes, supporting large-scale digitization of legacy archives. Despite these advances, dense annotations with small symbols and overlapping text on thin lines remain critical challenges, often requiring post-processing and text–symbol correlation to ensure reliable topology reconstruction.

A second group of contributions addresses electrical schematics. You Only Look Once (YOLO)-style detectors, adapted with domain-specific symbol sets, are shown in [21] to achieve high mAP for standard components and power connections, provided that class distributions are balanced and labeling is consistent. A convolutional front-end combined with graph traversal methods, introduced in [22], extracts grid parameters such as ratings and impedances directly from substation diagrams, supporting the derivation of computational network models. These methods demonstrate strong detection performance but remain sensitive to error propagation across recognition and OCR stages, which can compromise the physical coherence of reconstructed topologies.

A third research line focuses on diagram understanding through integrated detection, OCR, and layout analysis. A text-processing pipeline for mechanical drawings is presented in [23], where encoder models handle rotations and technical fonts, improving text–symbol association in title blocks and callouts, though residual errors may still affect downstream assemblers and limit portability across heterogeneous Computer-Aided Design (CAD) layers. From the perspective of semantics at graph level, ref. [24] introduces a method that models P&IDs as graphs, applying a graph neural network to classify functional line types and move beyond pixel-space features, yet requiring refined graph data rarely available in industrial contexts. In [25], a layered framework for electrical diagrams separates element, text, and connection layers, combining YOLO, OCR, and occlusion handling, before recomposing an electrical graph suitable for reasoning tasks.

Across these contributions, recent surveys such as [26] emphasize that although deep-learning-based methods have significantly advanced diagram digitization beyond traditional heuristic techniques, the field still faces major obstacles. Chief among these are the scarcity of publicly available annotated datasets, severe class imbalance, the lack of standardization across domains, and limited generalization of trained models to heterogeneous document types. Overall, existing approaches demonstrate clear advantages, in particular high precision in symbol and line detection, effective utilization of pre-trained architectures, and promising multimodal extensions, but also expose persistent limitations. Current frameworks remain largely domain-specific, depending on proprietary symbol libraries or manual post-processing, which restrict scalability. Moreover, most methods end with geometric or visual interpretation and fail to capture the underlying semantic relationships between components, resulting in partial or inconsistent topology reconstruction. Finally, the absence of integrated pipelines combining visual detection, text understanding, and knowledge representation limits their automation potential. These open challenges motivate the need for a comprehensive framework that unifies object detection, language-based reasoning, and ontology-driven KG modeling to achieve robust, semantically consistent digitization of engineering schematics. The following section explores how the integration of LLMs and KGs can address these issues by enabling structure-aware reasoning and semantic enrichment within automated industrial documentation workflows.

2.2. LLMs and Knowledge Graphs in Industrial Applications

Recent studies have explored multiple directions for integrating LLMs with KGs to improve retrieval, reasoning, and knowledge representation in industrial domains. A first group of contributions focuses on making Retrieval-Augmented Generation (RAG) frameworks more structure-aware. In [27], a document-structure KG is embedded within a GraphRAG pipeline to improve retrieval robustness and answer generation on technical corpora. This approach enhances contextual relevance and mitigates hallucinations in question–answering tasks. Similarly, ref. [28] introduces a reasoning-graph-based framework, CRP-RAG, which models reasoning processes explicitly to guide knowledge selection and dynamic planning during complex queries. These studies demonstrate how graph-based reasoning can improve the factual grounding and transparency of LLM responses. A second line of research investigates automated KG construction driven by LLMs. The framework proposed in [29] combines LLMs with multimodal modules to enrich incomplete graphs, leveraging text, audio, and visual data to address long-tail entities and improve reasoning efficiency. A comprehensive survey in [30] analyzes Knowledge Graph Embedding (KGE) techniques enhanced by LLMs, emphasizing how textual priors and symbolic rules can be fused to strengthen generalization and scalability. In operational contexts, ref. [31] presents a prompt-based LLM-driven approach for task-oriented KG construction, combining few-shot learning with semantic validation to reduce manual annotation and improve knowledge accuracy.

Beyond KG generation, several studies highlight their role as semantic infrastructures for Digital Twins (DTs) and plant-level decision support. For instance, ref. [32] proposes the PIDQA framework, which transforms P&IDs into queryable base graphs to enable natural-language interaction grounded in accurate entity and relationship extraction. In the same direction, ref. [33] develops a methodology to represent P&IDs as labeled property graphs using the Data Exchange in the Process Industry (DEXPI) standard (The DEXPI initiative defines a standardized, machine-readable XML format for the digital representation and exchange of P&IDs. It aims to ensure interoperability among computer-aided engineering tools and support the development of intelligent, data-centric process models. Further details are available at https://dexpi.org (accessed on 1 June 2026).) and integrates them with LLMs through graph-based RAG, enabling intuitive natural-language access to industrial schematics. Despite these significant advancements, existing approaches still present important limitations that motivate the framework proposed in this work. Current RAG-based systems exhibit high computational complexity and often depend on predefined ontologies, which restrict adaptability to new industrial domains. Automated KG construction methods driven by LLMs improve efficiency but frequently suffer from inconsistencies, limited control over semantic accuracy, and challenges in handling technical schematics that mix textual and graphical elements. Moreover, most multimodal or reasoning-graph frameworks remain largely theoretical or validated only on textual corpora, without addressing the peculiarities of engineering documentation such as symbol variability, spatial dependencies, and diagrammatic conventions.

Consequently, a unified and domain-oriented methodology is still lacking—one capable of connecting vision-based symbol detection, LLM-assisted semantic reasoning, and ontology-driven KG representation. The framework proposed in this paper addresses these gaps by combining advanced object detection, active learning, and semantic modeling into a coherent pipeline that can automatically interpret and transform P&IDs into structured, machine-interpretable knowledge suitable for industrial analysis and decision support. Table 1 summarizes the key features and limitations of the reviewed works. The comparison shows a clear evolution from vision-based object detection and OCR pipelines toward graph-oriented and LLM-assisted frameworks. While recent approaches improve semantic reasoning and automation, they remain fragmented, domain-dependent, and limited in scalability. The proposed framework overcomes these issues by integrating detection, text interpretation, and ontology-driven KG generation within a unified and extensible architecture.

Table 1. Comparative summary of the reviewed works on object detection and LLM–KG integration. Legend: ✓ = fully supported; ○ = partially supported; × = not supported.

Ref.	Obj. Detect.	Text/OCR	Graph	LLM/KG	Multimodal	Main Limitations and Remarks
[18]	✓	×	×	×	×	High accuracy in line recognition but lacks semantic mapping at device level.
[19]	✓	✓	×	×	×	No training data required; limited scalability to complex layouts.
[20]	✓	×	×	×	×	Robust line extraction; does not address symbol recognition or semantics.
[21]	✓	×	×	×	×	Effective for electrical diagrams; dataset imbalance and domain dependency.
[22]	✓	✓	✓	×	×	Graph-based parameter extraction; OCR propagation errors affect topology.
[23]	×	✓	×	○	✓	Focused on textual recognition; uses VL/LLM for checks (no KG); layout/font sensitivity.
[24]	×	×	✓	×	×	GNN-based line-type classification; requires pre-existing refined graph data.
[25]	✓	✓	✓	×	×	Layered detection and reasoning; high preprocessing cost and low semantic integration.
[27]	×	×	✓	✓	×	GraphRAG improves QA accuracy but depends on domain-specific ontologies.
[28]	×	×	✓	✓	×	Complex reasoning pipeline; validated only on textual data.
[29]	×	×	✓	✓	✓	Multimodal KG completion; limited semantic control and consistency.
[31]	×	×	✓	✓	×	Prompt-based KG construction; accuracy sensitive to few-shot design.
[32]	✓	✓	✓	○	×	Converts P&IDs into queryable graphs; needs entity/relationship normalization.
[33]	×	×	✓	✓	×	Graph-based RAG for P&IDs; prototype stage, high token cost, limited adaptability.
This work	✓	✓	✓	✓	○	Unified pipeline (detection → OCR → LLM reasoning → KG); assumes ontology/symbol availability; local LLM inference cost; spatial coordinates excluded by design.

3. Framework Architecture

P&IDs emerged as an evolution of Process Flow Diagrams (PFDs) extending their scope by incorporating detailed representations of control systems and physical connections between equipment. While PFDs illustrate general process operations, P&IDs explicitly represent critical components such as valves, measurement instruments, actuators, and other elements essential for plant monitoring and control. Accurate interpretation of these diagrams is vital across multiple domains, including new plant design, production process optimization, and maintenance planning. Engineers rely on P&IDs to validate equipment layout, develop automated control strategies, and ensure operational safety. However, their visual complexity poses significant challenges for automated analysis, as standardized symbols, connecting lines, and textual annotations are densely integrated within the diagrams. Computational processing of P&IDs requires advanced object detection techniques to accurately identify and classify diagram elements. When combined with knowledge representation models, the extracted information can be converted into structured and interpretable data, enabling deeper analysis of functional interconnections among system components.

The proposed approach aims to develop a scalable and adaptable system capable of interpreting P&ID diagrams and extracting structured information. The framework relies on a modular architecture that integrates advanced object detection, active learning, and Knowledge Graph generation techniques to enable the automated analysis of P&ID diagrams. The framework combines computer vision methodologies, knowledge representation and iterative learning, providing an innovative solution for automated analysis of complex technical documents in process engineering. As shown in Figure 1, the reference pipeline for P&IDs processing and KG generation consists of the following steps:

1. Pre-processing of P&ID images. As detailed in Section 3.1, this phase aims to improve the quality of input data into the recognition model removing unnecessary textual elements that might interfere with the object detection process and optimizing contrast and visual quality. These actions are essential to reduce noise in the image and improve the distinction between symbols, increasing recognition accuracy and minimizing false positives or negatives during automated analysis;
2. Object detection. Deep learning models (described in detail in Section 3.2) have been exploited to recognize and classify symbols in the diagrams. In particular, the YOLO model was adopted as reference framework for symbol recognition;
3. Feature extraction and image reconstruction. The framework integrates a Slicing Aided Hyper Inference (SAHI) [34] technique enhancing the detection of small objects, avoiding identification errors due to the complexity of the overall image;
4. LLM-based Knowledge Graph generation. After completing the extraction of symbols and the identification of related relationships, the framework converts all data into a structured representation. LLMs are leveraged to identify relationships between different P&ID objects and recognize recurring patterns that indicate functional or structural connections within the same plant, enabling deeper semantic understanding and contextual integration of extracted data. This semantic model allows the P&ID components to be organized into a KG consisting of nodes and relationships, enabling advanced analyses of technical correlations and plant operation patterns.

In particular, an Active Learning Cycle (ACL) [35] has been adopted in the third step to improve the performance of the recognition model. ACL is an iterative ML strategy in which a model is trained on a small initial labeled dataset and then progressively improved by selectively querying the most informative unlabeled samples for annotation. At each iteration, the model identifies uncertain or ambiguous instances, often using measures such as prediction entropy or margin confidence, and requests labels for these samples from an oracle (e.g., a human expert). The newly labeled data is then added to the training set, and the model is retrained, repeating the cycle until a performance target is reached or labeling resources are exhausted. The main benefit of this approach is that it significantly improves model performance while minimizing the amount of labeled data required. By focusing annotation efforts on the most informative samples, ACL enhances data efficiency, reduces labeling costs, and accelerates convergence in the ability to interpret P&ID patterns.

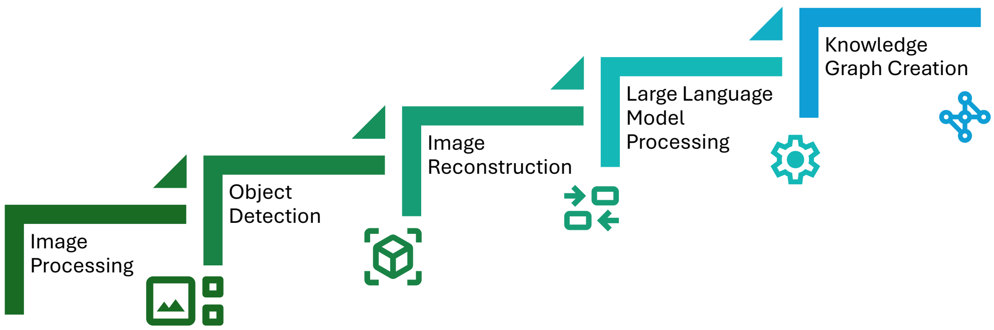

Figure 1. Reference pipeline for P&IDs processing and KG generation.

Each task is described in detail in the following sections, outlining the technical steps required to transform raw diagram images into structured and semantic-enhanced data.

3.1. Preprocessing of P&ID Schemas

P&ID diagrams are complex and detail-rich images and need to be processed properly before being passed to a neural network for automatic symbol detection. An example of P&ID is reported in Figure 2 including valves, pumps, sensors and other equipment. The analysis of P&ID diagrams aims to identify and accurately detect symbols representing several components of an industrial plant. Preprocessing P&ID diagrams ensures that the object detection model can operate with high accuracy. Preprocessing is also used to improve image quality, reduce noise and adapt the input format to the specifications of the deep learning model. This stage includes several operations: (i) normalizing images; (ii) removing superfluous text; (iii) correcting geometric distortions; (iv) increasing the visibility of technical symbols.

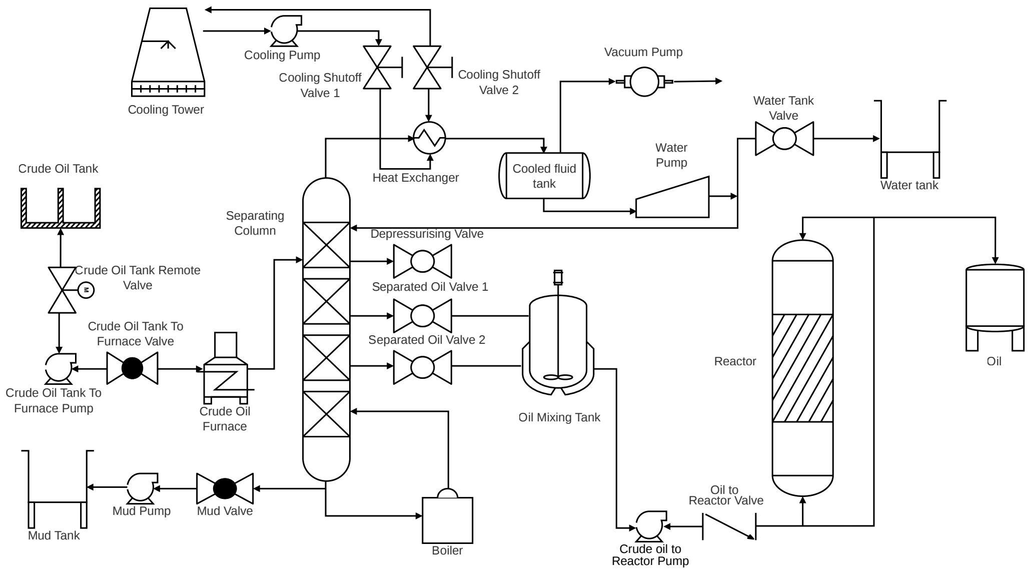

Figure 2. Example of P&ID schema.

The first step in the preprocessing stage is to apply grayscale conversion to the image. This reduces computational complexity by removing the color component, which is not essential for symbol detection. Additionally, grayscale conversion helps normalize the visual intensity of objects, improving pattern recognition even in the presence of strong contrasts across different parts of the diagram. Smoothing filters are also applied to reduce noise, which is a common issue in digital images [36]. Filters such as Gaussian Blur help to smooth out abrupt intensity variations that could otherwise compromise detection accuracy. P&ID diagrams typically include labels, annotations, and other textual elements that may interfere with the accurate detection of symbols. Consequently, these components can be removed during the preprocessing phase to ensure a reliable symbol recognition. Text removal is performed using OCR techniques to accurately locate and isolate areas containing text. An inpainting technique is applied to fill the identified text areas with the predominant background color, preserving the visual consistency of the diagram without altering its overall geometry. Finally, the preprocessing stage also includes normalization of image size. Since P&ID plots can vary significantly in dimensions and resolution, it is crucial to standardize all images to match the input requirements of the object detection model. This scaling ensures consistency across the dataset, facilitating effective model training and convergence.

3.2. Object Detection

YOLO is a series of computationally efficient models real-time object detection based on CNNs. It performs detection with a single image processing, ensuring fast and efficient analysis. The model splits the image into a grid of cells and, for each cell, generates predictions for objects. Every outcome has an associated probability and a bounding box indicating the location of the object. Each grid cell can contain multiple predictions, each associated with a class of object (e.g., valve, pump, sensor) and a related confidence probability. Object location accuracy is optimized using Intersection over Union (IoU), which measures the overlap between the bounding box prediction and the actual object location in the image. During model training, a specially designed dataset was used to represent a variety of symbols found in P&ID diagrams. This dataset includes symbols in different sizes, orientations and positions to simulate the complexity of real industrial diagrams. The error function used in the training process includes two main components: (i) localization loss, which measures the deviation between the predicted and actual bounding box location; (ii) classification loss, which evaluates the accuracy in assigning the correct class to the detected symbols. In particular, the YOLO model was trained using the backpropagation method, which allows the weights of the neural network to be updated while minimizing the error function. Furthermore, regularization methods were used to avoid overfitting the training data, as well as augmentation techniques, which modify some image features, such as rotations and translations, to improve the generalization ability of the model. Another key factor of the detection process is the use of the Non-Maximum Merging (NMM) filter. This filter is used to handle cases where the model produces multiple overlapping predictions for the same symbol in order to eliminate those with low confidence probability and retain only the most accurate prediction. The application of NMM improves detection quality by decreasing the chance of ambiguity in the location of objects. The final result is depicted in Figure 3, highlighting the enhanced image obtained respect to the original version in Figure 2. Through object detection, the YOLO model enables accurate mapping of symbols in P&ID diagrams, supporting the transition to the next steps of the framework.

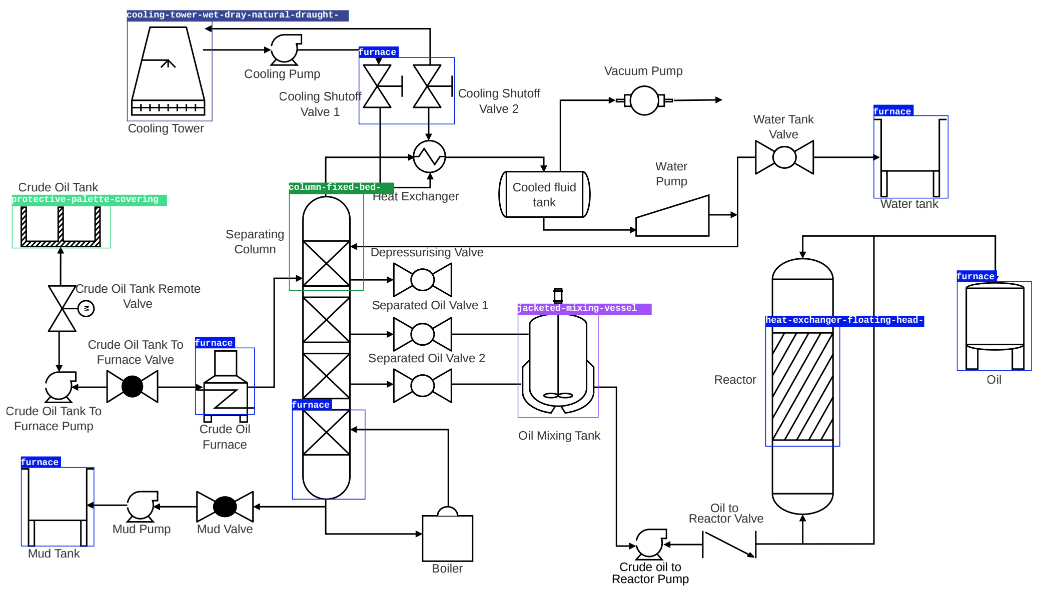

Figure 3. Example of object detection applied to a P&ID schema.

3.3. Feature Extraction and Image Reconstruction

The Feature Extraction phase in P&ID diagrams focuses on extracting distinctive features from the symbols in the images. Thus, they can be properly classified and mapped with their spatial relationships. A key aspect of this phase is symbol detection, which involves the use of advanced techniques such as Edge Detection. This is used to identify and define the shapes of technical symbols, which are often similar to each other. In the context of P&ID diagrams, symbols are frequently complex and overlapping, making them difficult to extract correctly unless treated with appropriate processing techniques. For example, valves, pumps, sensors, and other devices often exhibit similar geometric features (e.g., curved lines or rectangular contours) making it necessary to apply edge detection algorithms. Among the most commonly used methods of this type are the Canny [37] and Sobel [38] algorithms, which are designed to identify sharp boundaries between symbols and their surrounding regions. Thresholding techniques are then applied to obtain binary images, where symbol edges are clearly defined. Accuracy in feature extraction is further improved with techniques such as Slicing Aided Hyper Inference (SAHI), which divides the image into small analysis windows. This approach is particularly useful for detecting small symbols or those that, due to image density, might go undetected during global analysis. As shown in Figure 4, each window is processed separately improving the accuracy of symbol location. The dataset created for training these models includes a variety of symbols, with a total of around five hundred unique symbols divided into twenty-four major classes. The data augmentation approach includes techniques such as rotations and random image translations, thus increasing the variability of the data and helping improve the strength of the model in managing unfamiliar variants during training.

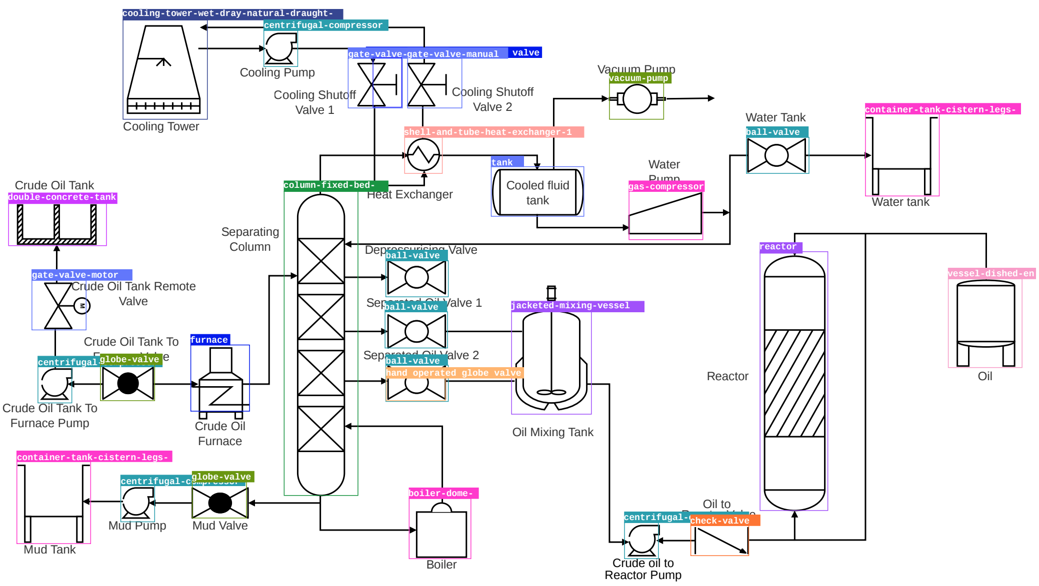

Figure 4. Enhanced P&ID schema obtained after the feature extraction procedure.

Finally the image reconstruction phase is carried out to identify and map the relationships among the various components. After feature extraction and symbol detection, the images were converted to grayscale or binary (black-and-white) formats (Figure 5) to reduce computational complexity and enhance the contrast between symbols and the background. This step enabled the reconstruction of the diagram structural layout, allowing the identification of spatial and functional relationships among components and the determination of main flow paths within the system.

https://doi.org/10.3390/bdcc10070210

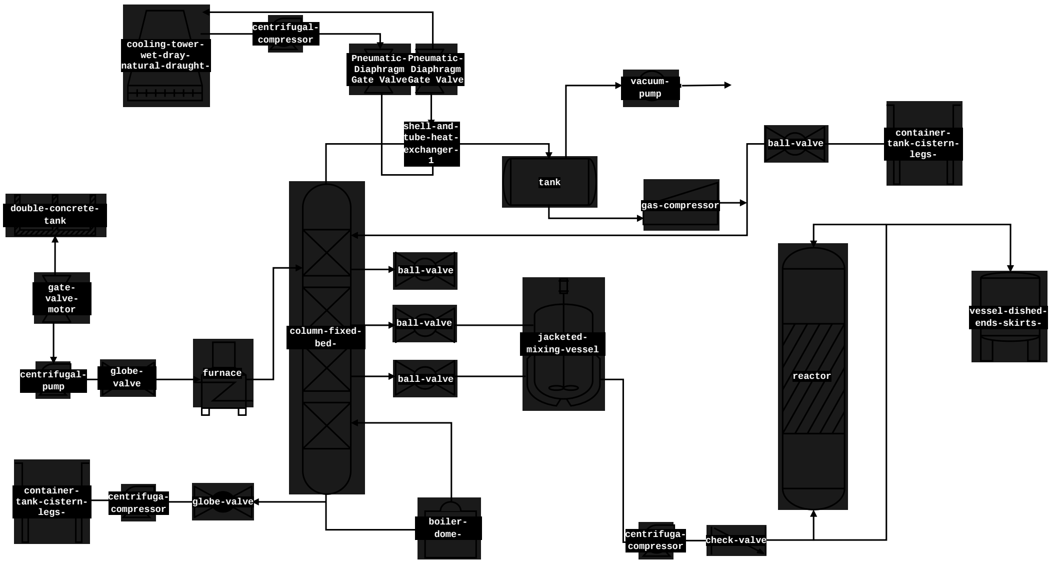

Figure 5. Example of black-and-white P&ID schema for image reconstruction.

3.4. LLM-Based Knowledge Graph Generation

The generation of a KG from P&ID diagrams concludes the process of digitizing and automating industrial plant management. This process transforms the information contained in the diagrams into a semantic-based data structure that clearly and consistently represents the interconnections between the various components and symbols identified in the diagram. Ontology design is the first step in the KG construction process and consists of defining a formal structure that represents concepts, entities and relationships in the specific domain of the P&ID diagram. This work adopts Linked Data (LD) principles [39] a set of guidelines for publishing and connecting structured data on the Web. LD strongly supports the use of pre-defined ontologies and standardized symbol sets when modeling KGs. By linking data across different sources, LD enables the creation of a distributed global graph of information, where related entities can be seamlessly discovered and integrated across domains. In the context of KGs derived from industrial diagrams, the exploitation of standardized ontologies ensure that the extracted knowledge is consistently structured and machine-interpretable, enabling interoperability across heterogeneous industrial systems and digital engineering platforms. In addition, ontologies provide a formal and semantically rich description of a specific reference domain by defining its entities, their properties, and the relationships that interconnect them. Unlike a simple enumeration of objects or terms, an ontology offers a structured and logically coherent model that captures the conceptual organization of knowledge within the domain. This formalization ensures semantic consistency, allowing information to be interpreted unambiguously across systems, and supports interoperability among heterogeneous data sources. Through the explicit definition of relationships and constraints, ontologies provide the foundation for knowledge integration, reasoning, and automated discovery within the broader ecosystem of interconnected data.

The Resource Description Framework (RDF) [40] has been adopted as the reference language for ontology definition. RDF provides a standardized and flexible data model for representing information through subject–predicate–object triples, enabling the explicit specification of entities, their properties, and relationships. Its graph-based structure ensures semantic interoperability, allowing the integration of heterogeneous data sources and facilitating reasoning and querying across interconnected datasets within the industrial knowledge domain. In particular, to represent the elements existing within the industrial domain, the following well-known RDF vocabularies have been adopted, each serving a specific modeling purpose:

- RDF Schema (RDFS) [41], used to define hierarchical relationships among entities, enabling the mapping of subclass-of relations within the ontology;
- SAREF4SYST (https://saref.etsi.org/saref4syst/ (accessed on 1 June 2026)), an extension of the Smart Applications REFerence ontology (SAREF) [42] designed to model different system typologies and their interconnections, supporting the representation of functional and structural dependencies between components;
- SAREF4BLDG (https://saref.etsi.org/saref4bldg/ (accessed on 1 June 2026)), an extension of SAREF tailored to the building domain, used to represent physical assets, spatial elements, and their relationships within built environments;
- SAREF4WATR (https://saref.etsi.org/saref4watr/ (accessed on 1 June 2026)), an extension of SAREF for modeling entities and processes in the water management domain, including water networks, treatment systems, and related infrastructure;
- SAREF4ENVI (https://saref.etsi.org/saref4envi/ (accessed on 1 June 2026)), an extension of SAREF for the environmental domain, supporting the representation of environmental conditions, storage systems, and related observations;
- PID vocabulary, a domain-specific vocabulary proposed to model elements unique to the reference scenario, providing specialized concepts and relationships relevant to P&ID.

The next step consists of schema mapping, which involves associating the information extracted from P&ID diagrams with the corresponding ontological structure. Although P&ID diagrams encode rich technical information, they remain primarily graphical artifacts and therefore cannot be directly interpreted by computer systems without an appropriate semantic framework. Schema mapping is the process of linking the symbols, labels, and visual connections extracted from a P&ID diagram to the concepts and relationships defined in a reference ontology, enabling semantic interpretation. To accomplish this mapping, the framework adopts both textual and semantic matching approaches. In the first phase, labels extracted from symbols are normalized to eliminate syntactic variations and to standardize technical terminology. Subsequently, the reference RDF vocabularies described above, containing standardized industrial concepts, are employed to perform an initial alignment between the extracted elements and their corresponding ontological entities. To refine the alignment between the identified symbols and the corresponding ontology entities, textual similarity techniques based on word embeddings and string matching are employed. In cases of ambiguity, the contextual information in which a symbol appears is used for resolving semantic conflicts.

The visual connections between symbols, obtained through the diagram analysis described in Section 3.3, are subsequently translated into semantic relationships that comply with the rules and constraints defined within the reference ontology. This process preserves the logical structure of the P&ID schema and ensures semantic coherence across the extracted data. LLMs are employed to further refine these relationships by identifying the semantic meaning associated with the connections among components, interpreting their functional roles and dependencies within the diagram. For each relationship, the LLM also generates a natural language description that captures its contextual relevance and technical significance. Gemini 1.5 Flash [43], an LLM developed by Google, was chosen for the semantic interpretation of connections acting as a solid framework in dealing with complex technical contexts. As reported in Table 2, the selection of the proposed model is motivated by its native multimodal architecture and its strong capability to jointly process text and visual inputs, combined with a very large context window. These characteristics enable efficient fusion of heterogeneous information sources and support scalable reasoning over complex and information-dense diagrams, such as flowcharts, technical schematics, and structured system representations, making the model particularly suitable for advanced diagram analysis tasks. The model is also able to interpret visual information and extract semantic relationships between symbols, accurately representing them in textual form. Furthermore, its ability to generate context-aware descriptions and incorporate technical details makes it a highly effective solution for P&ID diagram analysis, supporting the construction of a knowledge graph derived from visual data. The model has been configured with a maximum token limit per inference, where each token corresponds to a word or a sequence of words. This extended token capacity allows the model to manage detailed descriptions of symbol interactions, enabling the analysis of large and complex diagrams with high accuracy and contextual depth.

Table 2. Comparison of LLMs with focus on diagram understanding and multimodal capabilities.

Model	Diagram Understanding	Long Context	Key Strengths	Key Limitations
Gemini 1.5 Flash	Strong	Very large (up to millions of tokens in Gemini 1.5 family)	Fast inference, strong multimodal fusion, effective for diagram-to-text reasoning, flowcharts, and technical schematics	Slightly reduced reasoning depth compared to Gemini Pro/GPT-4o in edge-case logical reasoning
Gemini 1.5 Pro	Very strong	Extremely large (1–10 M token-scale context)	State-of-the-art visual reasoning, excellent for complex diagrams combined with long documents	Higher latency and computational cost
GPT-4o	Very strong	Large (smaller than Gemini 1.5 family)	Robust spatial reasoning, strong performance on scientific figures, UI diagrams, and mixed reasoning tasks	Less scalable for ultra-long multimodal documents compared to Gemini 1.5
Claude 3.5 Sonnet	Strong	Moderate context window	High-quality explanations of diagrams and strong summarization of visual content	Weaker spatial reasoning compared to GPT-4o and Gemini models
LLaVA 1.6	Moderate	Limited	Flexible and research-friendly, good for simple diagram understanding	Struggles with complex reasoning and long-context multimodal integration

The workflow begins with the creation of a structured prompt used as input to the model. This step tunes the model to understand the spatial and functional context of each symbol. The prompt is also enriched with information about the function of the symbols and their operational linkage. In this way, the model is able to recognize not only the symbols but also to identify their interactions within the diagram. The model's temperature is low to ensure that the model generates more deterministic responses, reducing the possibility of variability. Once the model receives the prompt, it processes the data and provides a textual description of the relationships between symbols. Usually, these descriptions are extremely detailed and cover the links between the various components, such as connection type and flow direction. Descriptions generated by the LLM model are used to refine and enhance the connection existing within the KG. As shown in Figure 6, the final outcome is a consistent and semantically enriched representation of the diagram in the form of a KG, which enables advanced analysis and the extraction of structured information from technical documentation. In this process, Gemini 1.5 Flash acts as an intermediate reasoning layer that bridges low-level visual perception and high-level semantic representation. It maps diagram elements to predefined ontological concepts and identifies functional relationships. This structured output can then be directly leveraged for LLM-driven KG construction, supporting downstream tasks such as querying, reasoning, and industrial process analysis.

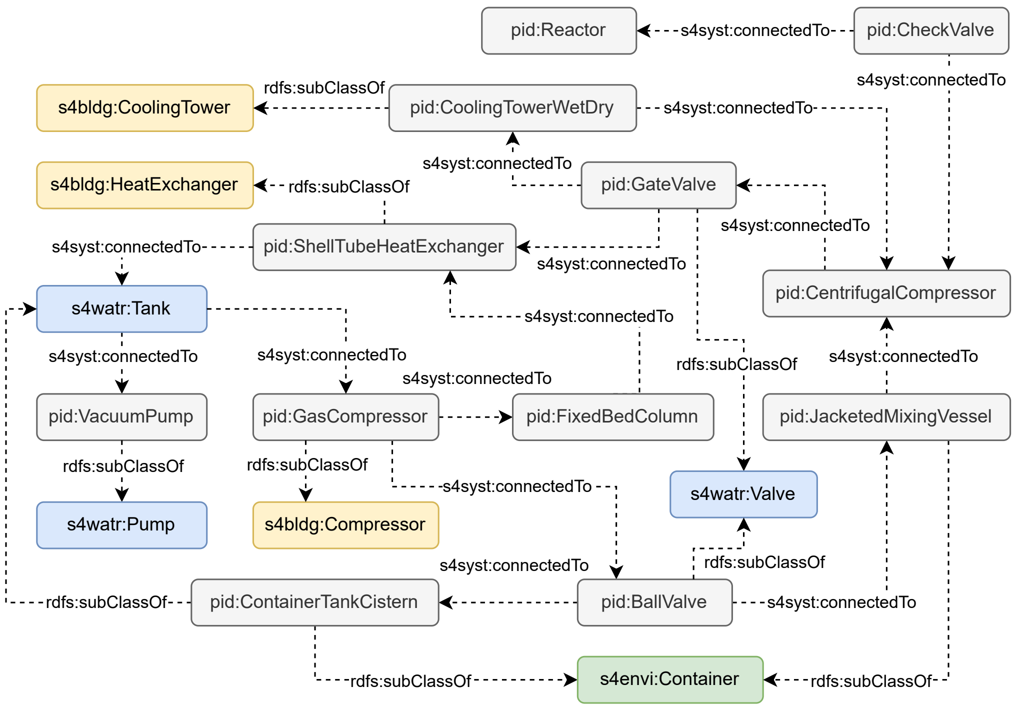

Figure 6. Example of a generated RDF-based Knowledge Graph.

Moreover, RDF triples composing the KG are stored in Neo4j (https://neo4j.com/ (accessed on 1 June 2026)), the graph database selected for experimentation. Each symbol detected in the diagram is represented as a node, while the interconnections between symbols are modeled as edges between nodes in the graph. This structured representation facilitates a visual and interactive exploration of the connections among the components of the P&ID, thereby supporting advanced analyses of plant structures and process flows. The system also includes an export functionality designed to model graph data in a structured and interoperable format, enabling subsequent processing through ML techniques or integration with interactive visualization tools. The export process supports main RDF serialization formats including JavaScript Object Notation for Linked Data (JSON-LD) [44], which provides a standardized and machine-readable representation of the KG, ensuring compatibility with semantic web technologies and data integration frameworks.

4. Experimental Results and Discussion

This section shows the results derived from the experiments conducted to analyze the effectiveness and accuracy of automatic P&ID pattern recognition and analysis algorithms. Details are also described regarding the preparation of the dataset, the methodology used to evaluate the models, and a discussion of the results.

4.1. Dataset Preparation

The dataset used for experiments consist of 126 P&ID diagrams collected from a variety of industrial sources, including plant manuals, real plant designs and diagrams generated by industrial CAD software (AutoCAD Plant 3D 2024, Autodesk, CA, USA, https://www.autodesk.com, accessed on 1 June 2026). The diagrams cover a wide range of plants and processes, including chemical plants, power plants, water treatment plants, and distribution networks. Collected diagrams vary in complexity, starting from basic diagrams with only a limited number of components, to very complex diagrams representing plants with an extensive network of piping, valves, pumps, control instrumentation and other equipment. Each diagram in the dataset was manually annotated with the goal of identifying and labeling individual diagram components. Annotations were made using a tagging system that links each symbol to a predefined class. In total, 5400 industrial symbols have been annotated and categorized into 24 main semantic classes. For each identified symbol a bounding box has been drawn, providing the detailed coordinates of the single component within the image. The complexity of the processed diagrams is heterogeneous, with an average of about 43 annotated components per diagram, while some diagrams contain more than 80 interconnected elements. This variability allowed the evaluation of the proposed approach under different levels of structural and semantic complexity typically encountered in real industrial environments.

Before model training, a pre-processing step was performed to normalize the visual characteristics of the images. This phase included resizing all images to a standard resolution, normalizing color and brightness values, and removing noise. Moreover, data augmentation techniques were applied to increase the variability of the dataset and improve the model's ability to generalize. The augmentation procedures included random rotations, translations, zooming, and variations in image contrast. Data augmentation was also used to simulate possible changes a diagram might undergo in real-world scenarios, such as tilting during scanning or changes in orientation when viewed on different devices. The final dataset was split into three subsets for training, validation, and testing. The training set accounted for 70% of the total dataset, while the validation and test sets covered 20% and 10%, respectively. This partitioning was performed randomly, while ensuring a balanced representation of all symbol types and components across the three sets. To prevent the models from overfitting to the training data, the test set was constructed to include diagrams that were not present in the training set. This allowed the simulation of real-world scenarios in which diagrams may originate from plants not previously encountered by the model.

4.2. Evaluation of Object Recognition Algorithms and KG Generation

The results obtained from training the P&ID symbol recognition model demonstrate strong and consistent performance across all evaluated metrics. As reported in Table 3, the model achieved an overall accuracy of 94.7%, confirming its ability to correctly classify the majority of detected P&ID symbols while minimizing false positives. The recall value of 94.6% confirms that the model can accurately detect most of the symbols contained in the diagrams, thus minimizing false negatives. This outcome is particularly relevant, considering the intrinsic variability of P&ID symbols diagrams in terms of size, orientations and positioning. A mean Average Precision (mAP) of 97.5% underscores the robustness and reliability of the proposed detection pipeline, indicating steady performance across varied scenarios involving symbols of various shapes, sizes, and positions within P&ID image diagrams. Such precision and recall values validate the model's suitability for industrial contexts, where symbol diversity and drafting inconsistencies often challenge automated analysis.

Table 3. Performance metrics for the object detection model.

Metrics	Value
mAP (mean Average Precision)	97.5%
Precision	94.7%
Recall	94.6%
Accuracy	94.7%

The evolution of the training process for the object detection model was evaluated by tracking both the loss functions and performance metrics across training epochs. As illustrated in Figure 7, the eight subplots collectively characterize the optimization behaviour typical of object detection frameworks. The training and validation losses, comprising box loss, class loss, and Distribution Focal Loss (DFL), decrease progressively over time. Box loss reflects improvements in bounding-box localization, class loss captures enhanced symbol classification accuracy, and DFL helps refine the regression of bounding-box distributions. The parallel reduction across training and validation losses indicates stable convergence and suggests that the model generalizes well without exhibiting signs of overfitting. Performance metrics further confirm these trends. Precision and recall rise consistently throughout training, reflecting the model increasing ability to correctly detect P&ID symbols while reducing missed detections and false positives. The growth and eventual stabilization of mAP@50 and mAP@50–95 demonstrate improved overall detection quality, covering both coarse and fine-grained localization thresholds. This opposite yet complementary trend demonstrates a consistent improvement in both localization and classification performance, confirming effective learning and robust generalization on the validation data. The continuous reduction in loss values throughout training indicates that the network successfully captured the discriminative visual patterns of each symbol, without showing signs of overfitting or underfitting. Moreover, the absence of oscillations or premature saturation in the curves highlights a stable optimization trajectory and an appropriate balance between the localization and classification tasks.

At a finer scale, the individual loss components exhibit complementary behaviours that reflect different aspects of model optimization. The box loss, which measures the accuracy of symbol localization within bounding boxes, decreases progressively, indicating that the model learns to position symbols with increasing spatial precision in complex layouts. The class loss, which evaluates the correctness of symbol categorization, follows a similar downward trend, demonstrating a gradual refinement of inter-class discrimination. Finally, the DFL loss, responsible for capturing fine-grained discrepancies in localization and classification, consistently declines, showing the model ability to reduce residual prediction errors and strengthen confidence in its detection outcomes. Collectively, the evolution of losses and metrics shows that the object detection model learns effectively, achieves stable convergence, and attains strong performance in recognizing and localizing symbols within P&ID diagrams.

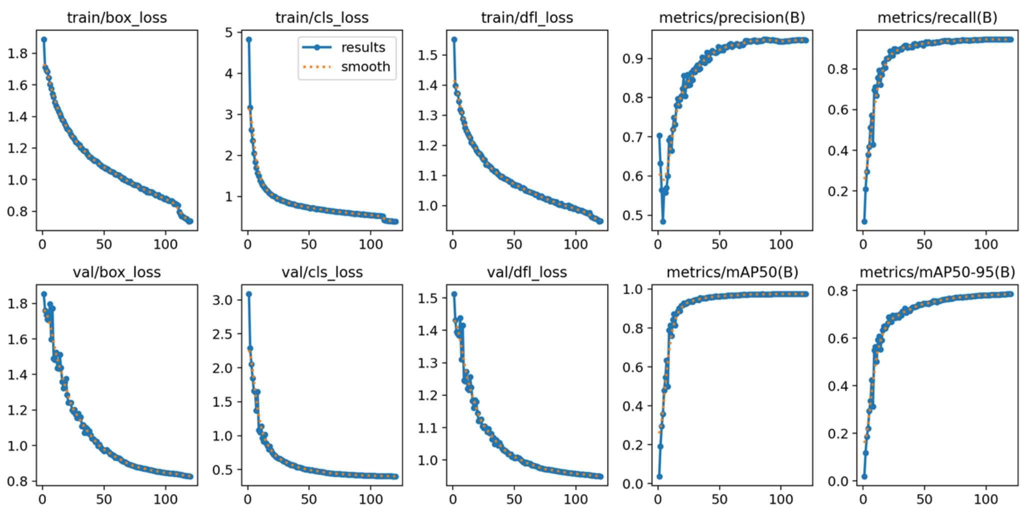

Figure 7. Training and validation metrics for the P&ID symbol recognition model.

The previous results demonstrate that the object detection stage provides a robust and reliable foundation for subsequent graph construction, ensuring that nearly all relevant entities are correctly identified. Successively, to assess the quality of the generated KG, a semantic validation analysis has been conducted focusing on both the correctness of the extracted relationships and the structural consistency of the graph representation. In particular, the evaluation considered the following aspects: semantic correctness of the generated triples; validity of the connections established between industrial components; correctness of the inferred flow direction; and absence of disconnected or orphan nodes within the graph structure. The analysis has been performed through a manual inspection of 2150 semantic triples generated by the proposed system, comparing the extracted graph representation against the original P&ID diagrams used as ground truth reference. The obtained results demonstrate a high level of semantic reliability of the generated knowledge graph. In particular, 91.8% of the generated triples were evaluated as semantically correct, confirming the effectiveness of the extraction and semantic mapping process. The correctness of the connections between industrial components reached 94.2%, indicating that the system was able to accurately reconstruct the structural relationships represented in the diagrams. The evaluation of flow-direction inference achieved an accuracy of 89.7%, highlighting the capability of the approach to capture directional dependencies and process flows within complex industrial schemes. Moreover, the KG generation process showed strong structural consistency, with 96.1% of the generated graphs free from disconnected nodes. The evaluation confirmed a strong correspondence between recognized entities and their semantic mappings proving that the proposed framework effectively captures both functional and contextual dependencies among components. Moreover, the consistency of these mappings across different diagrams highlights the ability of the ontology-based schema to generalize and maintain interoperability across heterogeneous industrial scenarios. The proposed KG generation pipeline preserves the full range of information extracted from the input schematics, with the sole exception of spatial coordinates. This omission is not a technical limitation but rather a deliberate design choice, as positional data are mainly intended for producing visually coherent diagrams and do not contribute semantically meaningful information to the underlying schema.

Finally, an execution time analysis has been conducted to evaluate the computational impact introduced by Google Gemini 1.5 Flash within the proposed pipeline. The average execution times observed during the experimentation are reported in Table 4. The experimental results show that the semantic inference stage performed represents the most computationally intensive component of the pipeline, accounting for nearly half of the total execution time. This behavior is expected due to the complexity of the multimodal reasoning task, which requires the joint interpretation of visual symbols, textual annotations, and structural relationships extracted from the industrial diagrams. Despite the additional latency introduced by the LLM module, the overall average execution time of approximately 13 s per diagram demonstrates that the proposed approach remains computationally feasible for semi-automated industrial knowledge graph generation workflows. Furthermore, the relatively limited overhead associated with KG construction and RDF triple generation confirms the efficiency of the downstream semantic processing stages. These results suggest that the integration of multimodal LLMs can significantly enhance semantic understanding capabilities while maintaining acceptable execution times for real-world industrial scenarios.

Overall, the generated KG provides a machine-interpretable representation of complex P&ID schematics and acts as a semantic backbone for downstream applications, including schema analysis, natural language-based manipulation, simulation input generation, and digital-twin integration. The semantic coherence achieved through ontology alignment, combined with the robustness of the detection pipeline, enables the construction of comprehensive and interpretable graph-based models that support advanced reasoning and decision-making processes in industrial contexts.

https://doi.org/10.3390/bdcc10070210

Table 4. Average processing time of the proposed pipeline.

Processing Task	Time (s)	% of Total Time
Image preprocessing	0.8	6.1
YOLO + SAHI object detection	2.9	22.0
Feature extraction and image reconstruction	1.1	8.3
LLM-based semantic inference	6.4	48.5
RDF triples generation	1.3	9.8
Neo4j graph generation	0.7	5.3
Total	13.2	100

4.3. Challenges and Limitations

The proposed framework represents an initial step toward addressing the digitalization challenges of industrial documentation through the integration of LLMs and automated KG generation. Although the achieved results confirm the technical soundness and feasibility of the approach, several challenges remain that may influence large-scale deployment and domain transferability. A first limitation concerns the absence of a standardized and comprehensive repository of industrial symbols and related ontologies for P&ID design. Existing standards, such as DIN 30600 [45] and ISO 14617-1 [46], only partially cover the diversity of graphical conventions adopted across industries. Consequently, current implementation still relies on manual symbol extraction and labeling to build training datasets and ontological mappings. Nonetheless, preliminary analyses suggest that this dependency can be mitigated, as the majority of symbols across industrial software suites share similar graphical structures and semantics.

Another challenging aspect concerns the reasoning capabilities of current LLMs when handling complex schematic representations. As discussed in Section 3.4, LLMs have proven effective in identifying and structuring relationships among detected symbols. However, this process focuses primarily on the detection of lines, connectors, and simple associations. Incorporating domain-specific engineering knowledge, effectively embedding specialized expertise within LLMs or dedicated Small Language Models (SLMs), could substantially enhance contextual understanding, enabling the reconstruction of partially degraded diagrams and the intelligent manipulation of schematic data. The interpretation of textual labels in densely populated or low-quality diagrams is another area for improvement. When schematics are derived from aged or scanned documents, text recognition and symbol association become increasingly error-prone due to noise, distortion, or the use of legacy notation systems. Addressing this issue will likely require hybrid strategies that combine optical character recognition with multimodal reasoning and alignment with historical datasets.

From an operational perspective, data privacy and security are also key considerations. Industrial documentation often contains sensitive process information, making it unsuitable for external, cloud-based inference. For this reason, local deployment of LLMs or inference performed within secured, on-premise environments emerges as the preferred strategy in the short term. Nevertheless, this approach raises legitimate concerns regarding computational costs and energy efficiency, particularly when running large models on dedicated Graphics Processing Unit (GPU) infrastructures. Ongoing comparative studies of private inference frameworks are beginning to assess trade-offs among model accuracy, latency, and energy consumption, suggesting that this area will remain an active focus of research. Finally, while the proposed framework is theoretically scalable to other industrial domains, such scalability relies on the availability of well-defined ontologies and pre-labeled symbol repositories. In contexts where these resources already exist or can be efficiently developed, the methodology can be rapidly adapted to new schematic types, e.g., electrical diagrams or mechanical layouts.

5. Conclusions

This paper presented a novel framework for the automated interpretation of industrial schematics through the integration of object detection techniques and KG generation supported by LLMs. The approach enables the transformation of static P&ID diagrams into structured, machine-interpretable representations, contributing to the broader goal of industrial digitalization and data-driven process optimization. The integration of LLMs has demonstrated to be effective in semantic structuring and contextual enrichment, although more work is required to improve their reasoning ability on complex diagrams and incorporate deeper domain-specific knowledge. Additional challenges include the lack of standardized industrial ontologies and symbol repositories, the need for robust multimodal reasoning over noisy or legacy documents, and the development of scalable privacy-preserving deployment strategies.

Experimental validation has also confirmed the robustness and reliability of the proposed pipeline. The object detection model achieved high precision, recall, and mean average precision values, with well-behaved convergence across training metrics, demonstrating its capability to accurately identify and classify industrial symbols under diverse conditions. The resulting KG preserved the completeness and semantic coherence of the original schematics, allowing the extracted information to be leveraged for downstream applications such as schema analysis, natural language-based interaction, simulation input generation, and digital twin integration.

Future research aims to expand the present pipeline towards achieving fully autonomous and domain-adaptive functionality. Promising areas include: (i) developing shared symbol repositories and unified ontological standards for PID and associated schematics; (ii) refining LLMs and SLMs to incorporate expert engineering knowledge for reasoning over incomplete or partially degraded documents; and (iii) optimizing on-premise inference to balance accuracy, latency, and energy efficiency in privacy-sensitive industrial settings. Future work may explore the exploitation of domain-specific fine-tuning and transfer learning techniques to facilitate broader adoption and cross-domain interoperability and the integration of specialized LLMs to automatically label previously unlabeled symbols. Finally, the proposed framework will be tested in other technical fields, e.g., electrical and mechanical schematics, to assess its scalability and compatibility with existing digital twin ecosystems.

Author Contributions: Conceptualization, F.L., V.F. and D.M.; methodology, F.L., F.S. and M.R.; software, F.L. and V.F.; validation, F.L. and V.F.; formal analysis, S.I., G.L., D.L. and F.S.; investigation, G.L., S.I. and D.L.; resources, F.L., V.F. and D.M.; data curation, F.L. and V.F.; writing—original draft preparation, F.L., V.F., S.I., D.L. and G.L.; writing—review and editing, D.M., F.S. and M.R.; visualization, F.L. and V.F.; supervision, F.L., G.L. and S.I.; project administration, D.M. and M.R.; funding acquisition, D.M. and M.R. All authors have read and agreed to the published version of the manuscript.

Funding: The research was supported by Digital Enterprise grant (grant number NIL6S28), co-funded by Lutech S.p.A. and the European Regional Development Fund for Apulia Region 2014/2020 Operating Program.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data available on request due to restrictions (e.g., privacy, legal, or ethical reasons).

Conflicts of Interest: F.L., V.F. and D.M. are employed by Lutech S.p.A.; S.I., G.L., F.S. and M.R. are partners in donkeyPower S.r.l. The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Matt, D.T.; Pedrini, G.; Bonfanti, A.; Orzes, G. Industrial digitalization. A systematic literature review and research agenda. *Eur. Manag. J.* **2023**, *41*, 47–78.
2. Emmert-Streib, F. From the digital data revolution toward a digital society: Pervasiveness of artificial intelligence. *Mach. Learn. Knowl. Extr.* **2021**, *3*, 284–298. [CrossRef]
3. Jan, Z.; Ahamed, F.; Mayer, W.; Patel, N.; Grossmann, G.; Stumptner, M.; Kuusk, A. Artificial intelligence for industry 4.0: Systematic review of applications, challenges, and opportunities. *Expert Syst. Appl.* **2023**, *216*, 119456.
4. Toghraei, M. *Piping and Instrumentation Diagram Development*; John Wiley & Sons: Hoboken, NJ, USA, 2019.
5. Moreno-García, C.F.; Elyan, E.; Jayne, C. New trends on digitisation of complex engineering drawings. *Neural Comput. Appl.* **2019**, *31*, 1695–1712.
6. Kaur, R.; Singh, S. A comprehensive review of object detection with deep learning. *Digit. Signal Process.* **2023**, *132*, 103812.
7. Yahya, M.; Breslin, J.G.; Ali, M.I. Semantic web and knowledge graphs for industry 4.0. *Appl. Sci.* **2021**, *11*, 5110. [CrossRef]
8. Li, X.; Lyu, M.; Wang, Z.; Chen, C.H.; Zheng, P. Exploiting knowledge graphs in industrial products and services: A survey of key aspects, challenges, and future perspectives. *Comput. Ind.* **2021**, *129*, 103449. [CrossRef]
9. Bilenchi, I.; Gramegna, F.; Loseto, G.; Ieva, S.; Scioscia, F.; Ruta, M. COWL: Pushing OWL 2 over the edge. *Internet Things* **2025**, *29*, 101439.
10. Tiddi, I.; Schlobach, S. Knowledge graphs as tools for explainable machine learning: A survey. *Artif. Intell.* **2022**, *302*, 103627. [CrossRef]
11. Van Assche, D.; Delva, T.; Haesendonck, G.; Heyvaert, P.; De Meester, B.; Dimou, A. Declarative RDF graph generation from heterogeneous (semi-) structured data: A systematic literature review. *J. Web Semant.* **2023**, *75*, 100753.
12. Pinto, A.; Ieva, S.; Tomasino, A.; Loseto, G.; Scioscia, F.; Ruta, M.; De Feudis, F. A Framework for Automatic Knowledge Base Generation from Observation Data Sets. In *Proceedings of the Current Trends in Web Engineering*; Casteleyn, S., Mikkonen, T., García Simón, A., Ko, I.Y., Loseto, G., Eds.; Springer: Cham, Switzerland, 2024; pp. 89–100. [CrossRef]
13. Zhong, L.; Wu, J.; Li, Q.; Peng, H.; Wu, X. A comprehensive survey on automatic knowledge graph construction. *ACM Comput. Surv.* **2023**, *56*, 1–62. [CrossRef]
14. Jin, B.; Liu, G.; Han, C.; Jiang, M.; Ji, H.; Han, J. Large language models on graphs: A comprehensive survey. *IEEE Trans. Knowl. Data Eng.* **2024**, *36*, 8622–8642. [CrossRef]
15. Ieva, S.; Bilenchi, I.; Gramegna, F.; Pinto, A.; Scioscia, F.; Ruta, M.; Loseto, G. Enhancing Last-Mile Logistics: AI-Driven Fleet Optimization, Mixed Reality, and Large Language Model Assistants for Warehouse Operations. *Sensors* **2025**, *25*, 2696. [PubMed]
16. Pan, S.; Luo, L.; Wang, Y.; Chen, C.; Wang, J.; Wu, X. Unifying large language models and knowledge graphs: A roadmap. *IEEE Trans. Knowl. Data Eng.* **2024**, *36*, 3580–3599. [CrossRef]
17. Ieva, S.; Loconte, D.; Loseto, G.; Ruta, M.; Scioscia, F.; Marche, D.; Notarnicola, M. A retrieval-augmented generation approach for data-driven energy infrastructure digital twins. *Smart Cities* **2024**, *7*, 3095–3120.
18. Moon, Y.; Lee, J.; Mun, D.; Lim, S. Deep Learning-Based Method to Recognize Line Objects and Flow Arrows from Image-Format Piping and Instrumentation Diagrams for Digitization. *Appl. Sci.* **2021**, *11*, 10054. [CrossRef]
19. Choi, J.; An, D.; Kang, D. Object Recognition Scheme for Digital Transformation in Marine Science and Engineering. *J. Mar. Sci. Eng.* **2023**, *11*, 1914. [CrossRef]
20. Yoochan, M.; Han, S.T.; Lee, J.; Mun, D. Extraction of line objects from piping and instrumentation diagrams using an improved continuous line detection algorithm. *J. Mech. Sci. Technol.* **2023**, *37*, 1959–1972. [CrossRef]
21. Bhanbhro, H.; Kwang Hooi, Y.; Kusakunniran, W.; Amur, Z.H. A Symbol Recognition System for Single-Line Diagrams Developed Using a Deep-Learning Approach. *Appl. Sci.* **2023**, *13*, 8816. [CrossRef]
22. Li, X.; Liu, X. Optimizing Parameter Extraction in Grid Information Models Based on Improved Convolutional Neural Networks. *Electronics* **2024**, *13*, 2717. [CrossRef]
23. Toro, J.V.; Tarkian, M. Optimizing Text Recognition in Mechanical Drawings: A Comprehensive Approach. *Machines* **2025**, *13*, 254. [CrossRef]
24. Kim, G.; Kim, B.C. Classification of Functional Types of Lines in P&IDs Using a Graph Neural Network. *IEEE Access* **2023**, *11*, 73680–73687. [CrossRef]
25. Cao, W.; Chen, Z.; Wu, C.; Li, T. A Layered Framework for Universal Extraction and Recognition of Electrical Diagrams. *Electronics* **2025**, *14*, 833. [CrossRef]

26. Jamieson, L.; Moreno-García, C.; Elyan, E. A review of deep learning methods for digitisation of complex documents and engineering diagrams. *Artif. Intell. Rev.* **2024**, *57*, 136. [CrossRef]
27. Knollmeyer, S.; Caymazer, O.; Grossmann, D. Document GraphRAG: Knowledge Graph Enhanced Retrieval Augmented Generation for Document Question Answering Within the Manufacturing Domain. *Electronics* **2025**, *14*, 2102. [CrossRef]
28. Xu, K.; Zhang, K.; Li, J.; Huang, W.; Wang, Y. CRP-RAG: A Retrieval-Augmented Generation Framework for Supporting Complex Logical Reasoning and Knowledge Planning. *Electronics* **2025**, *14*, 47. [CrossRef]
29. Yue, P.; Tang, H.; Li, W.; Zhang, W.; Yan, B. MLKGC: Large Language Models for Knowledge Graph Completion Under Multimodal Augmentation. *Mathematics* **2025**, *13*, 1463. [CrossRef]
30. Liu, B.; Fang, Y.; Xu, N.; Hou, S.; Li, X.; Li, Q. Large Language Models for Knowledge Graph Embedding: A Survey. *Mathematics* **2025**, *13*, 2244. [CrossRef]
31. Guo, C.; Liu, J.; Gao, W.; Lu, Z.; Li, Y.; Wang, C.; Yang, J. A Large Language Model Driven Knowledge Graph Construction Scheme for Semantic Communication. *Appl. Sci.* **2025**, *15*, 4575. [CrossRef]
32. Gupta, M.; Wei, C.; Czerniawski, T.; Eiris, R. PIDQA—Question Answering on Piping and Instrumentation Diagrams. *Mach. Learn. Knowl. Extr.* **2025**, *7*, 39. [CrossRef]
33. Alimin, A.A.; Goldstein, D.P.; Balhorn, L.S.; Schweidtmann, A.M. Talking like Piping and Instrumentation Diagrams (P&IDs). *Syst. Control Trans.* **2025**, *4*, 1676–1681. [CrossRef]
34. Akyon, F.C.; Altinuc, S.O.; Temizel, A. Slicing aided hyper inference and fine-tuning for small object detection. In *Proceedings of the 2022 IEEE International Conference on Image Processing (ICIP)*; IEEE: Piscataway, NJ, USA, 2022; pp. 966–970.
35. Garcia, D.; Carias, J.; Adão, T.; Jesus, R.; Cunha, A.; Magalhães, L.G. Ten years of active learning techniques and object detection: A systematic review. *Appl. Sci.* **2023**, *13*, 10667. [CrossRef]
36. Archana, R.; Jeevaraj, P.E. Deep learning models for digital image processing: A review. *Artif. Intell. Rev.* **2024**, *57*, 11. [CrossRef]
37. Agrawal, H.; Desai, K. Canny edge detection: A comprehensive review. *Int. J. Tech. Res. Sci.* **2024**, *9*, 27–35. [CrossRef]
38. Vincent, O.R.; Folorunso, O. A Descriptive Algorithm for Sobel Image Edge Detection. In *Proceedings of the Informing Science & IT Education Conference (InSITE)*; Informing Science Institute: Santa Rosa, CA, USA, 2009; Volume 40, pp. 97–107.
39. Bizer, C.; Heath, T.; Berners-Lee, T. Linked Data—The Story So Far. In *Linking the World's Information: Essays on Tim Berners-Lee's Invention of the World Wide Web*; Association for Computing Machinery (ACM): New York, NY, USA, 2023; pp. 115–143.
40. Raimond, Y.; Schreiber, G. RDF 1.1 Primer. W3C Note, W3C. 2014. Available online: https://www.w3.org/TR/2014/NOTE-rdf11-primer-20140624/ (accessed on 1 June 2026).
41. Guha, R.; Brickley, D. RDF Schema 1.1. W3C Recommendation, W3C. 2014. Available online: https://www.w3.org/TR/2014/REC-rdf-schema-20140225/ (accessed on 1 June 2026).
42. García-Castro, R.; Lefrançois, M.; Poveda-Villalón, M.; Daniele, L. The ETSI SAREF ontology for smart applications: A long path of development and evolution. In *Energy Smart Appliances: Applications, Methodologies, and Challenges*; John Wiley & Sons: Hoboken, NJ, USA, 2023; pp. 183–215.
43. Team, G.; Georgiev, P.; Lei, V.I.; Burnell, R.; Bai, L.; Gulati, A.; Tanzer, G.; Vincent, D.; Pan, Z.; Wang, S.; et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv* **2024**, arXiv:2403.05530.
44. Kellogg, G.; Longley, D.; Champin, P.A. JSON-LD 1.1. W3C Recommendation, W3C. 2020. Available online: https://www.w3.org/TR/2020/REC-json-ld11-20200716/ (accessed on 1 June 2026).
45. *DIN 30600:1985-11*; Graphical Symbols—Registration, Designation. German Institute for Standardisation: Berlin, Germany, 1985.
46. *ISO 14617-1:2025*; Graphical Symbols for Diagrams—Part 1: General Rules. Edition 3. International Organization for Standardization (ISO): Geneva, Switzerland, 2025.

https://doi.org/10.3390/bdcc10070210