Named Entity Extraction in Industrial Diagrams with Computer Vision and MLLMs

Jon Karl Weibull*, Filip Duarte dos Santos*, Jefferson Alves de Souza*, Denyson Tomaz de Lima†,
Melissa Lemos*‡, Marco Antonio Casanova*‡
*Instituto Tecgraf - PUC-Rio, Rio de Janeiro, RJ 22451-900, Brazil
Emails: {jweibull, filip, alvesjefferson, melissa}@tecgraf.puc-rio.br
†Petrobras, Rio de Janeiro, RJ 20231-030, Brazil
Email: denyson@petrobras.com.br
‡Department of Informatics - PUC-Rio, Rio de Janeiro, RJ 22451-900, Brazil
casanova@inf.puc-rio.br

***Abstract*—This paper presents a comprehensive evaluation of methods for extracting named entities (tags) from Piping and Instrumentation Diagrams (P&IDs), critical documents used throughout the lifecycle of industrial facilities. Efficient indexing and retrieval of information from such technical documentation play a key role in improving industrial project management. The paper proposes integrating deep neural networks for object detection with state-of-the-art Multimodal Large Language Models (MLLMs), specifically GPT-4o and LLaMA-3.2-Vision-90B. The methodology includes a preprocessing stage to isolate and downscale symbols before tag extraction. Experimental results on a challenging dataset of 30 real-world symbols show that MLLM-based approaches significantly outperform traditional OCR techniques (82%), achieving near-perfect accuracy in most cases. The approach demonstrates the robustness of MLLMs under noisy and degraded conditions, offering a practical solution for augmenting engineering document databases with reliable named entities. These findings highlight the potential of MLLMs to enable robust, high-precision tag text extraction, thereby streamlining the indexing and management of complex engineering documentation. All 30 test symbols used in this study are provided as supplementary material to ensure reproducibility.**

I. Introduction

Industrial projects generate a wide range of technical documents, often in large volumes, accumulated throughout the plant's lifecycle. Despite advancements in CAE (Computer-Aided Engineering) documentation software, many of these documents, especially in legacy projects, are still delivered by contractors exclusively as scanned or image-based PDF files.

Therefore, efficient indexing and retrieval of technical documents are essential to facilitate the management of industrial projects. A promising strategy in this context is the automated extraction of named entities, such as the identifiers or *tags*, present in piping and instrumentation diagrams (P&ID), which is the focus of this study.

P&IDs are fundamental for the design, construction, operation, and commissioning of industrial facilities, as they provide detailed information about the processes and systems involved. Figure 1 presents an example of a P&ID (partially obliterated for confidentiality reasons) illustrating some symbols and their respective identifiers.

In the context of P&IDs, tags are textual elements that indicate the type of element, its location, and an alphanumeric sequence to ensure uniqueness. Tags follow a formation rule standardized by norms, which may vary from company to company, but with one common characteristic: they must be unique. This feature makes tags excellent candidates for indexing the original P&IDs. For example, an application may allow users to click on a tag in a PDF file and navigate to the element with the same tag within a CAD/CAE 3D model, or vice versa. This functionality helps locate design elements, facilitating access to the necessary information throughout the project's lifecycle.

As illustrated in Figure 1, tags in P&IDs are often formatted as separate text elements, which may require post-processing to reconstruct the complete tag. Additionally, the text is typically enclosed within graphical shapes, such as circles or rectangles. These characteristics make P&IDs an interesting challenge domain for tag extraction.

The main contribution of the paper is to evaluate different strategies for tag extraction in P&IDs, combining the use of Multimodal Large Language Models (MLLMs) [1] with traditional computer vision methods, such as text localization and OCR (Optical Character Recognition) techniques. The strategies include a preprocessing stage with two main objectives: (1) to increase the likelihood of recovering all textual tags, which are often lost in images containing high-density information such as technical documents; and (2) to reduce the cost associated with using MLLMs by minimizing the amount of data sent to the second processing stage, both in terms of image size and prompt content.

The text identification approach was evaluated on a test set of 30 challenging real-world symbols, enabling a rigorous comparison against traditional OCR-based methods. The results show that MLLM-based strategies significantly enhance both accuracy and robustness, making them particularly well-suited for the text extraction stage of tag identification and for augmenting engineering document databases with reliable named entities.

Figure 2 presents the key steps of the proposed evaluation pipeline. As the literature review thoroughly covers a variety of

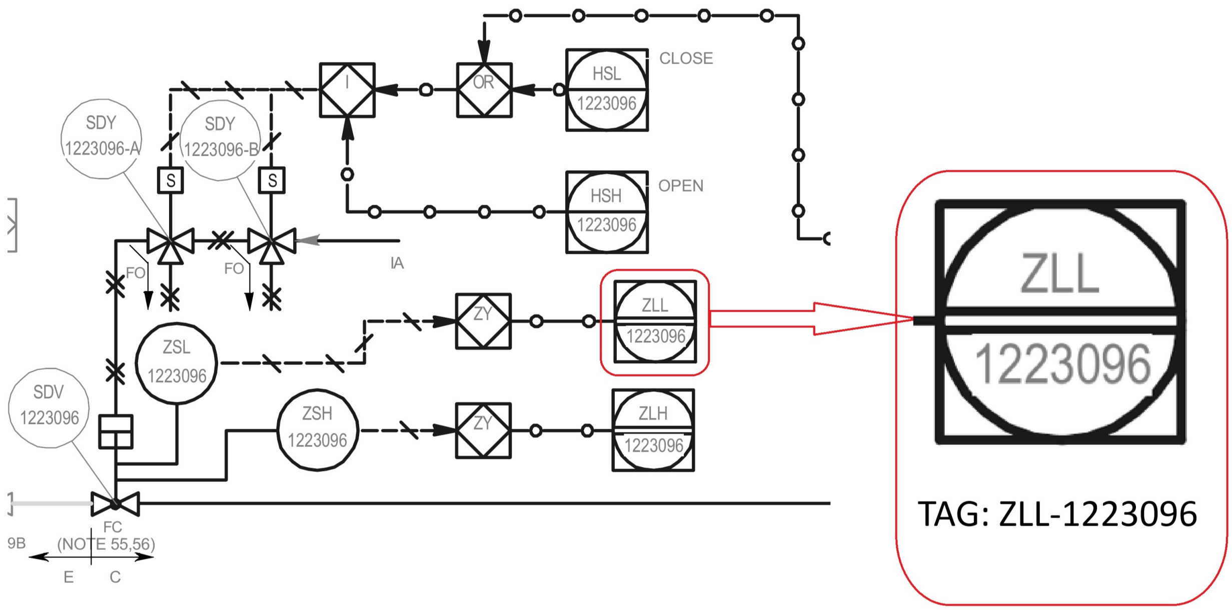

Fig. 1. Example of a P&ID diagram. Source: Prepared by the authors.

approaches for the computer vision stages, and given the space limitations of this paper, we focus our discussion primarily on the tag extraction evaluation. Consequently, details regarding the training and inference of the object detection network used for symbol cropping and downscaling are kept to a minimum.

This paper is organized into five sections. Section II presents related work. Section III describes the proposed method. Section IV presents and discusses the results. Finally, Section V provides the conclusions.

II. Related Work

This paper evaluates strategies for extracting key textual data from their respective engineering documents for future indexing. Inevitably, this work intersects with computer vision methods and symbol identification in engineering documents. In this field, several studies have addressed the challenge of extracting textual information from piping and instrumentation diagrams (P&IDs) using deep learning and image processing techniques. However, the focus seems to be much more oriented towards computer vision and semantic understanding of various diagrams and layouts rather than the extraction of key identifiers.

Yu et al. [2] proposed a method based on convolutional neural networks to recognize graphical components and texts in P&IDs in image format, achieving an accuracy of 83% in character recognition. Kim et al. [3] advanced this field by applying a deep learning architecture capable of handling high-density diagrams, using a combination of convolutional neural networks and attention mechanisms to accurately separate and recognize symbols and texts, achieving a precision of 92%.

Francois et al. [4] explored text detection combined with OCR post-correction techniques in technical documents, proposing a pipeline that uses an EAST model [5] for text detection and Tesseract [6] for extraction, achieving an accuracy of 82%.

Paliwal et al. [7] presented the Digitize-PID system, which uses deep learning-based techniques to extract textual and graphical entities from P&IDs, focusing on industrial applications. The work achieved a character recognition accuracy of 79%. Rahul et al. [8] complements this overview by proposing a complete system for the automatic extraction of information from P&IDs, integrating OCR (Tesseract) with structure recognition techniques (CTPN [9]) to interpret the relationship between texts and symbols.

These studies demonstrated that text extraction in P&IDs requires not only the application of traditional OCR but also integrated approaches that combine intelligent segmentation, deep learning, and contextual correction strategies to overcome the limitations imposed by the visual density and variability of the graphical elements in these documents.

Wang et al. [1] present a comprehensive survey on Multimodal Large Language Models (MLLMs), categorizing their applications across tasks involving natural language, vision, and audio. The survey offers a comparative analysis of various MLLMs in these domains. MLLMs are models that integrate image, language, and other modalities, often trained on massive datasets comprising trillions of tokens, and built with architectures containing hundreds of billions to trillions of parameters. This scale enables them to achieve advanced linguistic and contextual understanding. As such, multimodal models provide valuable references for designing specialized techniques and establishing baselines that dedicated solutions can aim to match or exceed.

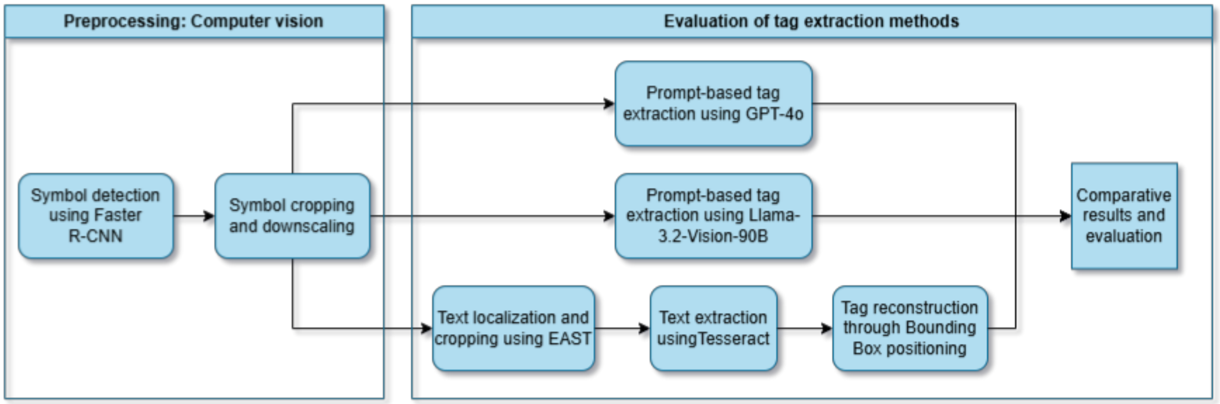

Fig. 2. Macro steps for extracting tags from P&IDs. Source: Prepared by the authors.

III. Strategies for Tag Extraction in P&IDs

Before proceeding with individual symbol extraction, an initial experiment was conducted using the MLLMs to directly read multiple tags from P&ID images containing several instrumentation symbols. However, this approach revealed significant limitations in accuracy, including errors in text recombination and incomplete tag identification. These shortcomings motivated the adoption of an individual analysis strategy, focusing on one symbol at a time to improve reliability and precision.

In outline, the proposed strategies for tag extraction in P&IDs have three major steps (see Figure 2):

1) Input Acquisition:
   a) Read a P&ID (in image format) from the database.
2) Preprocessing (Deep Learning Techniques):
   a) Detect symbols with tags.
   b) Extract and downscale each detected symbol.
3) Extract tags from symbols.
   a) Using ChatGPT-4o[1].
   b) Using Llama-3.2-Vision-90B.
   c) Using OCR (EAST + Tesseract).

The symbol detection step is responsible for identifying regions in the image that contain instrumentation symbols and forwarding only the delimited areas to subsequent stages. Various computer vision techniques were considered to implement this step, including traditional approaches such as template matching [10] and modern deep learning methods. In particular, the performance of two deep neural network architectures, Faster R-CNN [11] and SSD [12], was evaluated. On preliminary tests, among these, Faster R-CNN demonstrated superior performance, especially in handling densely packed and noisy diagrams.

YOLO-based detectors were considered during preliminary analysis. However, real-time inference was not a requirement for our use case, as tag extraction can be executed offline as part of background data processing. Therefore, YOLO's primary advantage, high-speed inference, was not essential for our objectives. Moreover, starting with YOLOv7, the official implementations are distributed under a GPL-3.0 license, which would require open-sourcing the entire pipeline or obtaining a commercial license for proprietary deployments. In contrast, Faster R-CNN offered a more permissive license and demonstrated superior detection accuracy on densely packed diagrams, making it a better fit for offline database augmentation scenarios, where accuracy is prioritized over inference speed.

The experiments were conducted using a dataset comprising P&ID images extracted from real-world documents. Both neural network models were fine-tuned on the same set of annotated images to evaluate the feasibility of accurately detecting and cropping complete symbols, yielding highly successful results. However, since symbol detection is well-covered in the existing literature and does not introduce novel challenges, our attention will shift toward the more distinctive task of tag extraction.

The proposed strategies diverge in the tag extraction step. A test set of instrumentation symbols was selected from real P&ID diagrams to assess the effectiveness of different text extraction techniques. These symbols were intentionally chosen to represent challenging real-world cases.

The baseline OCR strategy combines Tesseract and EAST. Tesseract is an optical character recognition (OCR) engine that converts non-editable text within images into machine-readable and searchable content [13], [14]. While it can be used independently for text recognition, Tesseract often struggles to accurately extract characters enclosed within graphical elements such as circles or lines, even in isolated symbols. To address this limitation, EAST (Efficient and Accurate Scene Text Detector) [5] is used as a preprocessing step. EAST is a deep neural network designed to detect text regions in complex scenes, such as street signs and billboards, and is particularly robust under noisy conditions. In this approach, EAST first identifies text regions, which are then passed to Tesseract

[1]After the paper was accepted, OpenAI announced GPT-5. Early experiments with GPT-5, using ChatGPT Pro, obtained the same results as GPT-4o, albeit the token consumption could not be verified.

for recognition. Additionally, the *whitelist* flag in Tesseract was enabled to reduce spurious character predictions. Finally, the extracted text must be reconstructed based on the spatial arrangement of EAST-detected bounding boxes relative to the original symbol.

The MLLM-based tag extraction strategies evaluated two models: GPT-4o [15] and Llama-3.2-Vision-90B [16], using the zero-shot learning [17] prompt illustrated in Figure 3. Interestingly, when prompted to extract text from images, MLLMs tend to apply OCR-like techniques by default. However, this behavior frequently resulted in incorrect outputs or complete failures. To overcome this limitation, our prompt explicitly instructed the models not to perform OCR but instead to rely on segmentation-based reasoning to identify and extract text. This prompt engineering led to significantly improved results, as reflected in the model outputs. It is also worth noting that the Llama-3.2-Vision-90B model was used in its Q4_K_M quantized configuration, as distributed by Ollama [18] for local deployment.

```
An isolated PID Symbol for an industrial
plant process design.
The Symbol is composed of a drawing and
text that represents an identifier tag.
Sometimes the tag might leak out of the
encircling image. The text might be in
angles such as 45 degrees or 90 degrees.
Don’t try to use OCR; use manual
segmentation to find the text and read
the tag in the image. Format your
response strictly as follows:
Tag: UpperText-LowerText.
```

Fig. 3. Prompt adopted for the experiments.

IV. Experiments

A. Test Dataset

The dataset used for training and evaluating the object detection networks consisted of over 150 P&ID images extracted from real-world engineering documents. Each image contained two sets of instrumentation symbols, manually annotated with bounding boxes and corresponding textual identifiers (tags), enabling supervised training for symbol detection. Ongoing work is focused on extending detection capabilities to a broader set of over 20 instrumentation symbols.

For the tag extraction experiments, a test subset of 30 individual symbols was manually selected. These samples were chosen to reflect a variety of challenging real-world conditions, including:

- Low resolution
- Visual noise and distortion
- Poor symbol formation
- Graphical interferences (e.g., overlaid lines and text)
- Ambiguous or degraded characters

These characteristics were intentionally selected to challenge the robustness of each extraction technique under adverse conditions. Symbols were carefully extracted and uniformly downscaled from the original P&IDs to ensure standardized inputs across all tested models. The complete set of symbols and their corresponding annotations is provided as supplementary material to support reproducibility[2].

B. Results

Table I presents the quantitative results of evaluating the three tag extraction strategies on the set of 30 isolated symbols. In this study, the traditional OCR-based approach reproduced the baseline accuracy of 82% reported in the literature, as shown in the OCR column of Table I. In contrast, the MLLM-based strategies significantly outperformed the OCR baseline across all evaluated metrics. GPT-4o and Llama-3.2-Vision-90B (in the Q4_K_M quantized configuration) exhibited excellent performance, with no fabricated characters and virtually no omission errors. Notably, GPT-4o achieved flawless extraction on the test dataset.

Figure 4(a) illustrates the only case in which Llama-3.2-Vision-90B failed, omitting the initial two characters ”II” from the tag. Despite this, the model successfully extracted the remaining alphanumeric portion, demonstrating its general reliability even in visually ambiguous scenarios.

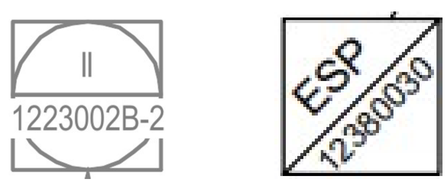

Fig. 4. (a) Prefix “II” in a tag; (b) Tag at a 45-degree angle. Source: Prepared by the authors.

When working with MLLMs, the token count is a critical factor, as it directly impacts the computational and financial cost of running the models in commercial applications. This is particularly relevant for image inputs, where the tokenization process follows a threshold-based behavior. Specifically, a combination of image size and visual complexity determines the number of tokens, which increases in discrete steps rather than linearly. In our experiments, since the prompt remains constant, it was observed that for symbol images up to 200×200 pixels, the input token cost remains fixed across both models.

Table II presents the token consumption observed for each model. The token counts were obtained directly from the metadata returned by the API for each inference request. Interestingly, the Llama-3.2-Vision-90B model consistently consumed approximately one-third of the total tokens used by the GPT-4o model, indicating a significant difference in processing cost between the two models.

[2]Available at `https://github.com/jweibull/suplementary_data_SIBGRAPI_2025.git`.

TABLE I
COMPARISON OF TEXT EXTRACTION RESULTS ON 30 SYMBOLS.

Metric	OCR	GPT-4o	Vision-90B
Correct characters extracted	290 (82.86%)	**333 (100%)**	331 (99.4%)
Missing characters	43 (12.91%)	**0 (0%)**	2 (0.6%)
Fabricated characters (false positives)	60 (17.14%)	**0 (0%)**	**0 (0%)**
Extracted / correct character count	350 / 333	**333 / 333**	331 / 333
Exact tags extracted	0 / 30	**30 / 30**	29 / 30

TABLE II
COMPARISON OF TOKEN CONSUMPTION FOR THE MLLMS.

MLLM	Input (Fixed)	Output (Average)	Total (Average)
GPT-4o	357	9.3	366.3
Llama-3.2-Vision-90B	107	11.37	118.37

Figure 5 illustrates the challenges of text extraction in low-resolution images with overlapping text, clearly showing that the MLLMs tested achieve superior results compared to the traditional OCR method in this critical scenario.

Recall that the baseline OCR strategy adopted Tesseract, with preprocessing for text localization and cropping using the EAST model. Without cropping, the success rate was practically zero. With cropping, valid results were obtained, although the extraction quality significantly degraded in images with low resolution or high noise levels. Furthermore, when the text overlapped with symbol lines, the technique completely failed. There was also high sensitivity to visually similar characters such as 'O' and '0', 'B' and '8', as well as 'l', '1', and 'I'.

An implicit advantage of using MLLMs is eliminating the post-processing step typically required to reconstruct tags based on the spatial arrangement of bounding boxes. Unlike traditional OCR pipelines, which require explicit merging of fragmented text components, MLLMs can return the complete, correctly formatted tag directly in response to the prompt. This is made possible by their contextual understanding of spatial and semantic relationships, even when the tag components appear at non-standard orientations, such as 45-degree angles, as shown in Figure 4(b). As a result, the overall pipeline is simplified, and extraction accuracy is improved in challenging visual scenarios.

Latency and inference time were not primary requirements in our proposed approach. The main objective of this work is to augment engineering document databases by creating reliable named entities (tags) to enhance search and retrieval capabilities. Since this process is designed to run as a background service, immediate real-time performance is not essential. Future work may explore optimizing inference speed for interactive or time-sensitive applications, but for the database augmentation scenario, accuracy and robustness were prioritized over latency.

For the proposed use case (indexing by identifiers), simply counting extracted characters may not be the most appropriate evaluation metric. In this application, the identifier must be extracted completely and accurately, without alterations, to avoid a subsequent post-processing step, which could be

Fig. 5. (a) Image containing the TAG; (b) Tesseract; (c) GPT-4o; (d) Llama-3.2-Vision-90B. Source: Prepared by the authors.

complex. In the evaluated examples, where the text was often blended with graphical elements of the symbols, Tesseract could not correctly extract a single tag. In contrast, the MLLM-based strategies correctly identified the characters and returned the identifiers already composed in their final format, which substantially simplifies the positional reconstruction required by traditional approaches.

V. CONCLUSION

Accurately extracting textual identifiers from industrial technical documents, particularly Piping and Instrumentation Diagrams (P&IDs), is a challenging task due to the frequent overlap of graphical and textual elements, variable image quality, and complex layouts. This study evaluated strategies combining traditional computer vision techniques with Multimodal Large Language Model (MLLM) capabilities, specifically GPT-4o and LLaMA-3.2-Vision-90B.

The results demonstrate that MLLM-based approaches significantly outperform conventional OCR-based strategies in extracting complete and accurate tags, especially in visually degraded or noisy scenarios. While traditional OCR (Tesseract + EAST) reproduced the expected 82% accuracy baseline, it failed entirely in extracting full tags in more complex cases. In contrast, GPT-4o achieved perfect extraction, and LLaMA-3.2-Vision-90B performed similarly, missing only two characters on a single tag.

Importantly, the strong performance of the MLLM-based approaches was enabled by careful preprocessing steps, including symbol detection, cropping, and prompt engineering, which instructed the models to avoid default OCR behaviors in favor of segmentation-based reasoning. This underscores that MLLMs do not function effectively in isolation; their success depends on well-structured and context-aware inputs.

In addition, understanding the threshold-based behavior of token generation in MLLMs is essential for optimizing their use. Token count is not linearly proportional to image size or complexity, and being aware of these thresholds helps define the practical limits for image resolution and prompt design. Finally, the substantial reduction in token usage observed with LLaMA-3.2-Vision-90B, consuming approximately one-third of the tokens used by GPT-4o, highlights its potential as a cost-effective alternative for local deployments.

While latency and inference time were not primary requirements in this study, given that the proposed method is designed for offline database augmentation rather than real-time applications, these factors remain relevant for future deployment scenarios. Accuracy and robustness were prioritized in the current work, and future research will investigate optimization strategies to reduce inference time for interactive or time-critical use cases.

All 30 test symbols and their annotations are provided as supplementary material to support reproducibility. The extracted tags generated by this approach can serve as reliable named entities for indexing, facilitating more efficient retrieval and management of engineering documentation throughout the industrial project lifecycle. Future efforts will also focus on scaling this pipeline to larger datasets and broader symbol classes while maintaining high accuracy and manageable computational costs.

Acknowledgment

This work was partly funded by FAPERJ under grants E-26/204.322/2024, by CNPq under grant 305.587/2021-8, and by the Petrobras Research, Development, and Innovation Program.

References

[1] J. Wang, H. Jiang, Y. Liu, C. Ma, X. Zhang, Y. Pan, M. Liu, P. Gu, S. Xia, W. Li, Y. Zhang, Z. Wu, Z. Liu, T. Zhong, B. Ge, T. Zhang, N. Qiang, X. Hu, X. Jiang, X. Zhang, W. Zhang, D. Shen, T. Liu, and S. Zhang, "A comprehensive review of multimodal large language models: Performance and challenges across different tasks," 2024. [Online]. Available: https://arxiv.org/abs/2408.01319

[2] E.-S. Yu, J.-M. Cha, T. Lee, J. Kim, and D. Mun, "Features recognition from piping and instrumentation diagrams in image format using a deep learning network," *Energies*, vol. 12, no. 23, p. 4425, 2019.

[3] H. Kim, W. Lee, M. Kim, Y. Moon, T. Lee, M. Cho, and D. Mun, "Deep-learning-based recognition of symbols and texts at an industrially applicable level from images of high-density piping and instrumentation diagrams," *Expert Systems with Applications*, vol. 183, p. 115337, 2021.

[4] M. Francois, V. Eglin, and M. Biou, "Text detection and post-ocr correction in engineering documents," in *International Workshop on Document Analysis Systems*. Springer, 2022, pp. 726–740.

[5] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang, "East: an efficient and accurate scene text detector," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 5551–5560.

[6] R. Smith, "An overview of the tesseract ocr engine," in *Ninth international conference on document analysis and recognition (ICDAR 2007)*, vol. 2. IEEE, 2007, pp. 629–633.

[7] S. Paliwal, A. Jain, M. Sharma, and L. Vig, "Digitize-pid: Automatic digitization of piping and instrumentation diagrams," in *Trends and Applications in Knowledge Discovery and Data Mining: PAKDD 2021 Workshops, WSPA, MLMEIN, SDPRA, DARAI, and AI4EPT, Delhi, India, May 11, 2021 Proceedings 25*. Springer, 2021, pp. 168–180.

[8] R. Rahul, S. Paliwal, M. Sharma, and L. Vig, "Automatic information extraction from piping and instrumentation diagrams," in *Proceedings of the 8th International Conference on Pattern Recognition Applications and Methods - ICPRAM*, INSTICC. SciTePress, 2019, pp. 163–172.

[9] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao, "Detecting text in natural image with connectionist text proposal network," in *Computer vision–ECCV 2016: 14th European conference, amsterdam, the netherlands, October 11-14, 2016, proceedings, part VIII 14*. Springer, 2016, pp. 56–72.

[10] R. Brunelli, *Template Matching Techniques in Computer Vision: Theory and Practice*. John Wiley & Sons, 2009. [Online]. Available: https://www.wiley.com/en-us/Template+Matching+Techniques+in+Computer+Vision%3A+Theory+and+Practice-p-9780470517062

[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," in *Proceedings of the 29th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'15. Cambridge, MA, USA: MIT Press, 2015, p. 91–99.

[12] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Computer Vision – ECCV 2016*, ser. Lecture Notes in Computer Science, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., vol. 9905. Springer, 2016, pp. 21–37. [Online]. Available: https://doi.org/10.1007/978-3-319-46448-0_2

[13] R. Smith, "An overview of the tesseract ocr engine," in *ICDAR '07: Proceedings of the Ninth International Conference on Document Analysis and Recognition*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 629–633. [Online]. Available: https://storage.googleapis.com/pub-tools-public-publication-data/pdf/33418.pdf

[14] R. W. Smith, "The extraction and recognition of text from multimedia document images." *DISS. ABST. INT. PT. B- SCI. & ENG., 1988,*, vol. 49, no. 4, 1988.

[15] OpenAI, "Introducing gpt-4o and more tools to chatgpt free users," https://openai.com/index/gpt-4o-and-more-tools-to-chatgpt-free/, 2024, [Online]. Accessed: 27 May 2025.

[16] Meta, "Llama 3.2: Revolutionizing edge ai and vision with open, customizable models," https://ai.meta.com/blog/llama-3-2-connect-2024-vision-edge-mobile-devices/, 2024, [Online]. Accessed: 27 May 2025.

[17] X. Sun, J. Gu, and H. Sun, "Research progress of zero-shot learning," *Applied Intelligence*, vol. 51, no. 5, pp. 3600–3614, 2021, accepted: 07 November 2020; Published online: 16 November 2020; Issue date: June 2021. [Online]. Available: https://doi.org/10.1007/s10489-020-02075-7

[18] Ollama, "Ollama: Run large language models locally," https://ollama.com/, 2024, [Online]. Accessed: 2 June 2025.