XLVI CILAMCE
VITÓRIA - ES - BRAZIL - 2025
XLVI IBERO-LATIN AMERICAN CONGRESS ON COMPUTATIONAL METHODS IN ENGENEERING

Automating Fire Protection Engineering Plan Analysis: Deep Learning for Symbol Detection

Pedro D. Boina[1], Dr. Luiz A. Pinto[1]

[1]*Dep. of Control and Automation Engineering, Instituto Federal do Espírito Santo, Brazil*
Serra, 29166-630, Espírito Santo, Brazil
pedrodalviboina@gmail.com, luiz.pt@ifes.edu.br

Abstract. This study presents a deep learning-based approach for the automatic detection of graphic symbols in fire protection engineering plans, aiming to automate the inspection of fire protection system designs. Fire protection plays a vital role in safeguarding lives, property, and the environment against the destructive effects of fires. In this context, engineering plans serve as critical tools for ensuring that buildings and facilities are equipped with appropriate safety measures. The manual verification of these plans is a highly visual task that demands sustained concentration, is prone to human error, and consumes a significant amount of time from public agents responsible for technical review. The integration of computer vision techniques with machine learning algorithms can reduce the analysts' workload while enhancing the efficiency and reliability of the inspection process. A dataset comprising 1,000 real images of fire protection plans submitted to the Military Fire Brigade of the State of Espírito Santo, Brazil, was used. A total of 5,267 graphic symbols were manually annotated, distributed across 12 imbalanced classes representing common elements in fire protection system layouts. A YOLOv11 convolutional neural network was trained to detect these symbols in real time. A test set was created by randomly extracting 10% of the original dataset. Model evaluation was performed using a stratified k-fold cross-validation technique, yielding a mean Average Precision at IoU 0.5 (mAP@50) of 93.7%. After training, the model was tested on test set, achieving an mAP@50 of 97.7%. To assess practical applicability, two software prototypes were developed for the automatic detection of symbols in engineering plans. Although detection time metrics were not included in this phase, the study demonstrates the feasibility of automating fire protection project verification using AI-based methods. Future work includes expanding the dataset, addressing class imbalance, and exploring alternative neural network architectures to enhance model robustness and generalization capabilities. This research contributes to the advancement of intelligent systems in fire protection engineering, offering a promising foundation for the integration of computer vision tools into regulatory workflows and technical project evaluation.

Keywords: Fire protection engineering, Deep learning, Computer vision, Object detection, Engineering drawings automation

1 Introduction

The technical review of fire protection engineering plans is a mandatory step in obtaining permits and licenses from public authorities. It plays a key role in ensuring that buildings and facilities comply with the minimum safety standards established in applicable regulations. In Brazil, this responsibility is assigned to the State Military Fire Departments, which analyze architectural floor plans and descriptive memorials with a focus on fire and panic safety measures [1].

This verification process is carried out through visual inspection of graphical elements present in the plans, such as fire extinguishers, hydrants, smoke detectors, manual call points, and emergency lights. Despite ongoing digitalization of administrative procedures, this inspection is still performed manually by technical officers, requiring continuous attention, detailed regulatory knowledge, and professional experience. As the demand for new construction increases and technical staff remains limited, this manual process becomes a bottleneck for fire safety services, leading to delays and potential inconsistencies in evaluations [2].

Partial or full automation of the plan review process thus emerges as a promising strategy to enhance efficiency, reduce processing times, and standardize technical decisions. With recent advances in computer vision and deep learning, it has become feasible to develop models capable of detecting and classifying graphical symbols in fire protection plans in real time — even under adverse conditions such as scale variation, overlapping elements,

and graphical noise [3, 4].

Among the most established object detection architectures in the literature, the YOLO (You Only Look Once) family stands out for its combination of high accuracy and low inference latency [5]. In this study, we propose an approach based on a recent YOLO version, YOLOv11, to automate part of the technical inspection workflow in fire protection engineering.

The dataset used consists of 1,000 real images extracted from architectural plans submitted to the Military Fire Brigade of the State of Espírito Santo (CBMES), totaling 5,267 manually annotated symbols distributed across 12 standardized classes according to the ABNT NBR 14100 technical standard [6]. A supervised training process was applied, and the model was evaluated using stratified cross-validation and an independent test set.

To assess practical feasibility, a software prototypes were developed to simulate an AI-assisted technical inspection environment, with a focus on automatically detecting fire protection symbols in digital plan files. The proposed system aims not only to accelerate technical verification but also to enhance traceability and reliability in decision-making by public regulatory agencies.

The next section reviews the related work on the automatic detection of graphical symbols in technical drawings, highlighting key challenges and approaches adopted in the literature.

2 State of the art

The automatic detection of graphical symbols in technical documents is a growing research field, driven by recent advances in computer vision and deep learning techniques. Applications in this area have been explored in several domains, including civil, electrical, and architectural engineering projects, traffic sign recognition, instrumentation diagrams, and emergency plans. These studies share a common goal: to reduce the manual workload associated with symbol interpretation and to increase the speed and accuracy of technical analysis processes.

Among the most widely adopted object detection architectures, the YOLO (You Only Look Once) family stands out for its ability to identify multiple object classes in real time with high performance. In [7], Yang and Zhang compared YOLOv3 and YOLOv4 in traffic sign detection and found that the newer version achieved significantly better results, though still showing limitations in detecting small-sized targets.

In the engineering domain, Rezvanifar et al. [3] applied YOLOv2 to detect symbols in digital architectural floor plans using a tile-based approach to handle graphical complexity and low intra-class similarity. The method proved effective for detecting 12 symbol classes, reaching a mAP@50 of 89.3%.

Akgul and Kutucu [8] proposed a system for automated placement of electrical symbols in 2D architectural plans using the Inception-v3 network with transfer learning, achieving an accuracy of 97.05% in classifying electrical components. Corneli et al. [9] developed a custom YOLO model to detect emergency equipment (such as extinguishers and emergency signs) in photographic images, obtaining mAP values ranging from 16% to 99%, depending on the dataset used.

More recently, Jamieson et al. [4] assessed the performance of YOLOv7 and Faster R-CNN in detecting symbols in high-resolution industrial plant drawings to automate tasks such as material takeoffs. The models demonstrated practical viability, with mAP@50 above 79% and significant time savings compared to manual analysis.

In the specific context of fire protection plans, Hassaan et al. [10] combined segmentation, color filtering, and object detection techniques using Faster R-CNN to locate emergency symbols in risk maps. Additionally, Schönfelder et al. [11] integrated computer vision and BIM models, using YOLOv8 and Keypoint R-CNN to automatically extract fire safety equipment from evacuation plans.

These studies confirm the increasing relevance of convolutional neural networks in the automatic detection of graphical symbols in technical environments. However, research focused specifically on CAD-based fire protection engineering plans remains limited. This highlights the originality and relevance of the present work, which uses real data and addresses the integration of deep learning with public-sector technical review systems.

3 Methodology

This section presents the procedures adopted for dataset development, object detection model training, and performance evaluation using cross-validation techniques. The objective is to establish a robust technical foundation for implementing an automated solution capable of identifying, in real time, the graphical symbols present in fire protection engineering plans.

3.1 Dataset Development

The dataset used in this study was built from 1,000 real images extracted from fire protection plans submitted to the Military Fire Brigade of the State of Espírito Santo (CBMES), Brazil. The original DWG files were visualized using Autodesk TrueView® 2019 software, and screenshots were taken with the Windows Snipping Tool, applying variations in zoom, scale, and background color (black and white) to increase dataset diversity.

The images were saved in JPG format, with 96 dpi resolution and RGB color mode. Each image contains at least one graphical symbol corresponding to a fire safety feature. Symbol annotation was performed manually using the LabelImg tool, with bounding boxes created according to the YOLO annotation format. The symbol classes followed the standardized representations defined in the ABNT NBR 14100 technical standard, totaling 12 categories, including extinguishers, hydrants, detectors, alarms, and emergency lighting, Figure 1.

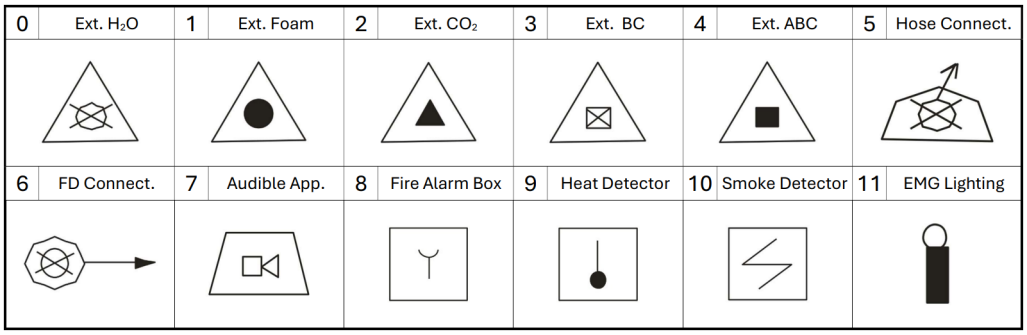


Figure 1. Fire protection graphic symbols selected for this work.

Upon completion of the annotation process, the dataset included 1,000 JPG images and 1,000 corresponding TXT label files, totaling 5,267 annotated instances. The distribution among classes was unbalanced, reflecting the actual frequency of these symbols in real-world projects. The average number of objects per image was 5.27. Table 1 displays how instances are distributed by class within the original dataset and its training and test splits.

Table 1. Instance count per class in the training (900), test (100), and full (1000) datasets

Class-ID	Class Name	Train (n=900)	Test (n=100)	Original (n=1000)
0	Ext_H2O	303	37	340
1	Ext_Foam	106	6	112
2	Ext_CO2	230	17	247
3	Ext_DryPowder_BC	491	66	557
4	Ext_DryPowder_ABC	328	37	365
5	Hose_Connection	381	56	437
6	FD_Connection	178	25	203
7	Audible_Notif_Appliance	333	45	378
8	Manual_Fire_Alarm_Box	354	47	401
9	Heat_Detector	145	31	176
10	Smoke_Detector	486	110	596
11	Emg_Lighting	1277	178	1455
Total		**4612**	**655**	**5267**

3.2 Preprocessing and Data Splitting

To ensure reliable model evaluation, the dataset was split into training, validation, and testing subsets. Initially, 10% of the images were randomly set aside as a final test set and were not used during model training. The remaining 90% were used in the training and validation stages using stratified k-fold cross-validation with k=5.

This approach ensures that each image is used once as a validation sample, while maintaining the class proportions in each fold. This is particularly important in imbalanced datasets. The shuffling and partitioning were performed using the scikit-learn library.

3.3 Model Architecture and Training Configuration

The YOLOv11 architecture, released by Ultralytics, was used in this study. This version introduces several improvements over previous generations, including a deeper backbone network, adaptive anchor boxes, advanced regularization techniques, and optimized augmentation for small object detection.

The "medium" YOLOv11 model was selected to balance detection capability and computational cost. Input images were resized to 640 × 640 pixels, and the training process ran for 500 epochs. A Tesla A100 GPU with CUDA 12.4 support was used for training.

The loss function included penalties for bounding box localization, class confidence, and classification accuracy. The initial learning rate was set to 0.01 with an adaptive decay policy. During training, some data augmentation techniques were applied, including horizontal and vertical flipping and rotation up to 30 degrees.

3.4 Evaluation Metrics

The model was evaluated using standard object detection metrics: Precision, Recall, mAP@50, and mAP@[0.5:0.95], following the COCO benchmark. The F1-Score was also calculated to provide a harmonic mean of precision and recall. In addition, normalized confusion matrices were generated to analyze class-level errors and identify patterns of misclassification between similar symbols.

Metrics

To assess the performance of the trained network, the main object detection metrics were used: *Precision* (P), *Recall* (R), *F1-score*, and *mean Average Precision* (mAP).

Precision (P) measures the proportion of correct detections among all detections made:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{1}$$

Recall (R) represents the proportion of ground-truth objects that were correctly detected:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2}$$

F1-score is the harmonic mean of precision and recall, offering a balanced measure:

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

mAP@50 refers to the mean of average precision values computed for each class, using a detection threshold where the *Intersection over Union (IoU)* is at least 0.5:

$$\text{mAP@50} = \frac{1}{N} \sum_{i=1}^{N} AP_i(\text{IoU} \geq 0.5) \tag{4}$$

mAP@50–95 is a stricter metric that averages the AP values calculated at IoU thresholds ranging from 0.50 to 0.95 with a step of 0.05. All these metrics were extracted from the validation reports generated by YOLOv11 during the stratified k-fold training and evaluation process.

The results obtained in each fold of the stratified cross-validation were averaged to summarize the model's performance. This approach ensures a more comprehensive evaluation across the dataset and reduces the influence of potential biases associated with specific data partitions.

4 RESULTS AND DISCUSSION

After training and validating the YOLOv11 model with the proposed dataset, performance metrics were obtained through stratified 5-fold cross-validation, as described in Section III. A final evaluation was also conducted using a test set composed of 100 independent images. The results shown in this section are fictional and serve as a structural placeholder for future updates.

4.1 Cross-Validation Results

The results indicate that the model performed well in detecting graphical symbols, with balanced precision and recall. The mean mAP@50 reached 93.7%, demonstrating that the model is capable of correctly identifying most of the symbols, even with an imbalanced class distribution and despite the limited number of training images. The F1-score demonstrates a balance between precision and recall. Table 2 shows the results obtained from training across the five folds using stratified k-fold cross-validation.

Table 2. Overall performance metrics (all classes) per fold

k-Fold	Validation images	Instances	Precision	Recall	F1-Score	mAP@50	mAP@50-95
Fold 0	180	819	0.929	0.901	0.915	0.947	0.793
Fold 1	180	939	0.922	0.875	0.898	0.934	0.765
Fold 2	180	954	0.929	0.884	0.906	0.950	0.795
Fold 3	180	1054	0.911	0.820	0.863	0.911	0.746
Fold 4	180	846	0.942	0.874	0.907	0.942	0.784
Mean	**180**	**922.4**	**0.927**	**0.871**	**0.898**	**0.937**	**0.777**

The mAP@50 values across the five folds exhibited low variability, with a standard deviation of 0.014, corresponding to 1.49% relative to the mean. This low dispersion indicates consistent model performance across different data partitions, suggesting a high level of generalization and stability in the detection results.

4.2 Per-Class Results

Table 3 details the mAP@50 and the other metrics for each of the 12 symbol classes during cross-validation training step.

Table 3. Average performance metrics per class across 5-fold cross-validation

ID	Class	Precision	Recall	F1-Score	mAP@50	mAP@50-95
0	Ext_H2O	0.839	0.815	0.827	0.888	0.763
1	Ext_Foam	0.936	0.893	0.914	0.927	0.801
2	Ext_CO2	0.948	0.932	0.940	0.951	0.815
3	Ext_DryPowder_BC	0.877	0.890	0.883	0.943	0.802
4	Ext_DryPowder_ABC	0.939	0.875	0.906	0.960	0.813
5	Hose_Connection	0.937	0.903	0.919	0.965	0.800
6	FD_Connection	0.918	0.917	0.918	0.939	0.790
7	Audible_Notif_Appliance	0.943	0.813	0.873	0.900	0.722
8	Manual_Fire_Alarm_Box	0.961	0.806	0.876	0.902	0.740
9	Heat_Detector	0.936	0.917	0.926	0.960	0.782
10	Smoke_Detector	0.941	0.916	0.928	0.976	0.835
11	Emg_Lighting	0.942	0.792	0.861	0.933	0.694
	Average	**0.927**	**0.871**	**0.898**	**0.937**	**0.777**

Overall, most classes achieved high detection performance, regardless of their frequency in the dataset. Notably, some less frequent classes such as Ext_Foam and Ext_CO2 reached precision and recall values close to

or equal to 1.0, indicating that the model was able to learn their visual features effectively. On the other hand, variability in F1-scores and mAP@50-95 for certain classes suggests that intra-class visual similarity and symbol complexity, rather than sample count alone, may influence detection performance.

4.3 Final Test Results

On the final test set (100 images not used during training or validation), the model demonstrated strong generalization capability, as shown in Table 4, reaching 97.7% of mAP@50.

Table 4. Detection performance per class on the 100-image test set including F1-score

Class	Images	Instances	Precision	Recall	F1-score	mAP@50	mAP@50-95
Ext_H2O	26	37	0.974	1.000	0.987	0.995	0.901
Ext_Foam	3	6	0.862	1.000	0.926	0.995	0.661
Ext_CO2	14	17	0.925	1.000	0.961	0.995	0.872
Ext_DryPowder_BC	40	66	0.911	1.000	0.954	0.995	0.840
Ext_DryPowder_ABC	18	37	0.982	0.973	0.977	0.993	0.810
Hose_Connection	41	56	0.899	0.954	0.926	0.968	0.794
FD_Connection	22	25	0.992	1.000	0.996	0.995	0.837
Audible_Notif_Appliance	35	45	0.972	0.775	0.863	0.938	0.720
Manual_Fire_Alarm_Box	36	47	0.907	0.872	0.889	0.960	0.764
Heat_Detector	8	31	0.967	0.934	0.950	0.984	0.793
Smoke_Detector	20	110	0.938	0.967	0.952	0.987	0.810
Emg_Lighting	48	178	0.927	0.826	0.874	0.925	0.677
All Classes	100	655	**0.938**	**0.942**	**0.940**	**0.977**	**0.790**

Figure 2 shows the confusion matrix obtained from the evaluation on the test dataset. The model demonstrates strong class-wise discrimination, particularly for frequent classes such as Emg_Lighting and Smoke_Detector, which exhibit high true positive counts and minimal confusion. Misclassifications are generally sparse, with a few noticeable overlaps between visually similar classes, such as Ext_CO2 and Ext_DryPowder_BC, or between Manual_Fire_Alarm_Box and nearby classes. The presence of some false positives labeled as background also highlights the challenge of symbol localization under complex plan conditions.

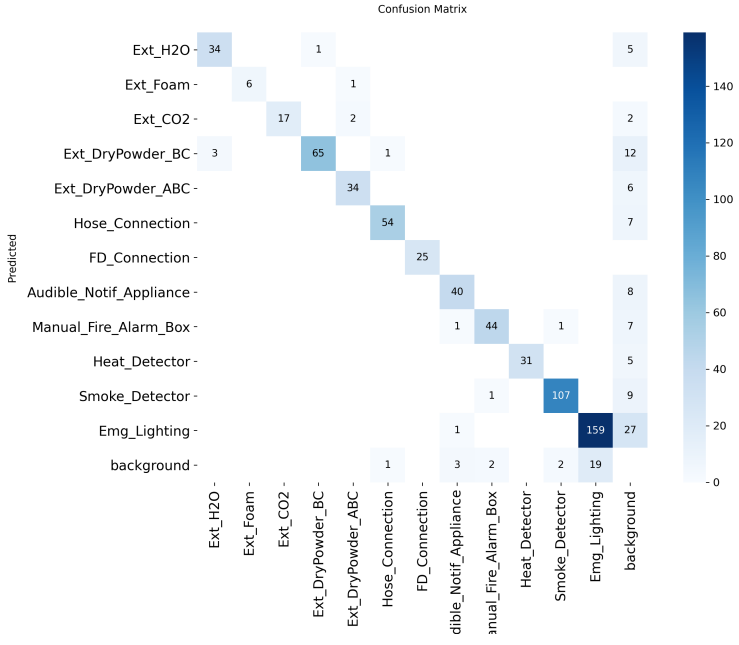


Figure 2. Confusion matrix obtained from model evaluation on the test set.

The high accuracy in the test set confirms the model's robustness and suitability for real-world applications. Notably, no synthetic data was used, which highlights the feasibility of applying the method using real technical documentation.

A deeper examination of misclassified cases revealed that most confusions occurred between visually similar extinguisher types and alarm devices with overlapping graphic elements. Future research could explore feature-level visualization to better understand model attention and support targeted improvements.

Despite the extensive manual annotation process, the dataset presents class imbalance reflecting real project frequencies. Future dataset updates should include targeted augmentation of underrepresented classes and the introduction of new symbol categories to enhance representativeness.

4.4 Discussion on real application

YOLOv11 proved effective for the proposed task, combining fast inference with high accuracy. Some challenges remain, particularly in detecting small or low-contrast. Real-time screen capture tests were conducted for symbol detection in fire protection plans (Figure 3), yielding promising results that indicate the model's potential for practical application.Future work will focus on dataset expansion, class balancing techniques, and synthetic data augmentation.

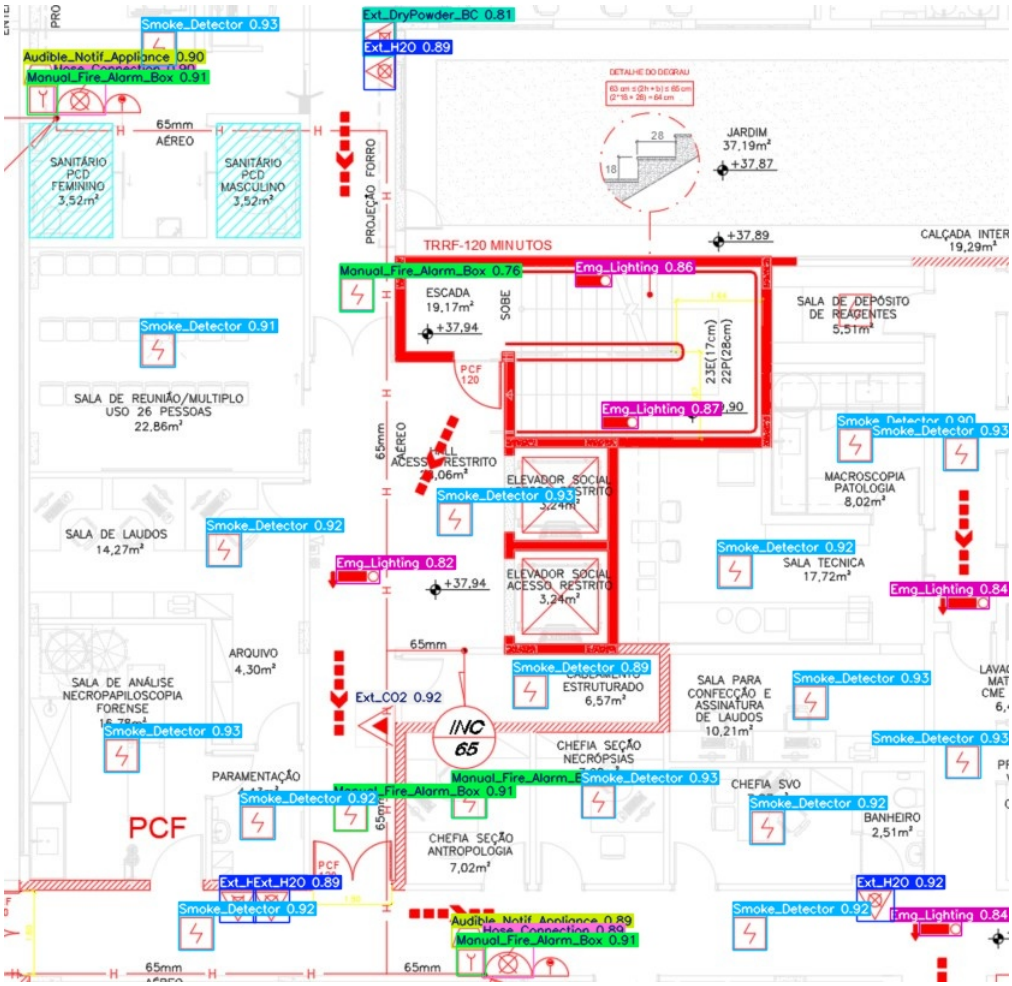


Figure 3. Real world test example.

An analysis of confusion matrices (not included here due to space constraints) revealed that the most frequent misclassifications occurred between similar extinguisher types, which share graphical characteristics. This underscores the need to improve inter-class discrimination through more annotated examples and further tuning of training hyperparameters. Although detection accuracy was high, future studies should include quantitative evaluation of computational efficiency, including inference speed (FPS) and memory usage, to assess scalability for large-scale plan repositories and real-time deployment in public institutions.

5 Conclusion

This study presented a deep learning-based approach for the partial automation of technical reviews of fire protection engineering plans, focusing on the automatic detection of standardized graphical symbols in digital architectural drawings. The proposed methodology involved the development of a real-world dataset based on projects submitted to the Military Fire Brigade of the State of Espírito Santo (CBMES), the training of a YOLOv11 object detection model, and performance evaluation through stratified cross-validation and an independent test set.

The results demonstrated that the YOLOv11 architecture is capable of accurately identifying fire protection symbols required by regulation, even in an imbalanced dataset with significant visual variability. The model achieved an mAP@50 of 93.7% in cross-validation and 97.7% on the test set, indicating strong potential for real-world deployment in technical review environments.

In addition to model training, the study included the development of software prototypes aimed at integrating automatic detection into the workflow of project inspections. Although still in early stages, these tools demonstrate the feasibility of incorporating artificial intelligence into the routine tasks of public agents, contributing to improved efficiency, standardization, and traceability in technical decision-making.

Future work will focus on expanding the dataset with new classes and annotated images, applying class balancing techniques (such as oversampling and synthetic data generation), and testing alternative architectures such as DETR, RT-DETR, and multimodal segmentation models. Additionally, user studies involving technical analysts are recommended to evaluate system usability and the real-world impact of such tools in the decision-making process.

The results reinforce the role of computer vision techniques as strategic tools for modernizing public regulatory services, particularly in the field of fire safety engineering.

Authorship statement. The authors hereby confirm that they are the sole liable persons responsible for the authorship of this work, and that all material that has been herein included as part of the present paper is either the property (and authorship) of the authors, or has the permission of the owners to be included here.

References

[1] Corpo de Bombeiros Militar do Estado do Espírito Santo. *Norma Técnica 01 – Parte 2: Apresentação de Projeto Técnico*. CBMES. `https://www.cbmes.es.gov.br/legislacoes-em-vigor`, 2015.

[2] A. I. Seito, A. A. Gill, F. D. Pannoni, R. Ono, S. B. d. Silva, U. D. Carlo, and V. P. e. Silva. *A Segurança Contra Incêndio no Brasil*. Projeto, São Paulo, 2008.

[3] A. Rezvanifar, M. Cote, and A. B. Albu. Symbol spotting on digital architectural floor plans using a deep learning-based framework. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 2419–2428, 2020.

[4] L. Jamieson, C. F. Moreno-García, and E. Elyan. Towards fully automated processing and analysis of construction diagrams: Ai-powered symbol detection. *International Journal on Document Analysis and Recognition*, 2024.

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

[6] Associação Brasileira de Normas Técnicas. *NBR 14100: Proteção contra incêndio — Símbolos gráficos para projeto*. ABNT, Rio de Janeiro, 1998.

[7] W. Yang and W. Zhang. Real-time traffic signs detection based on yolo network model. In *2020 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, pp. 354–357, 2020.

[8] B. Akgul and H. Kutucu. An automated system for electrical power symbol placement in electrical plan drawing. *Automatika*, vol. 63, n. 1, pp. 78–89, 2022.

[9] A. Corneli, H. Fadli, L. Tagliabue, A. Brischetto, D. Pagano, and M. Spirito. Training of yolo neural network for the detection of fire emergency assets. In *Proceedings of the 37th International Symposium on Automation and Robotics in Construction (ISARC)*, pp. 836–843. IAARC, 2020.

[10] M. Hassaan, A. Asif, A. Hassan, M. T. Sadiq, and S. H. Lee. Emergency floor plan digitization using machine learning. *Sensors*, vol. 23, n. 19, pp. 8344, 2023.

[11] S. Schönfelder, C. Kattwinkel, J. Bremer, and M. König. Enriching bim models with fire safety equipment using keypoint-based symbol detection in escape plans. *Automation in Construction*, vol. 162, pp. 105382, 2024.