From Keyframes to Narrative: A Multi-Stage AI Pipeline for Scene Understanding Using Object Detection and Large Language Models

Valentín Calzada-Ledesma[1][0000−0001−9328−971X], Victor Castillo-Olivetto[1], Emmanuel Márquez-Robles[1], Carlos Orozco-Solis[2], and Faustino Neri-Larios[2]

[1] Tecnológico Nacional de México/ITS de Purísima del Rincón, Purísima del Rincón Guanajuato 36425, México
[2] Emerald Digital, León Guanajuato 37530, México

Abstract. This paper presents a novel multi-stage framework for automatic video understanding and storytelling. The proposed system first identifies keyframes using a temporal segmentation strategy to extract the most informative moments from the input video. Each selected keyframe is then processed through the YOLO object detection model to identify and classify relevant objects within the scene. Finally, a large language model (GPT-4o-mini) is employed to generate coherent and context-aware natural language descriptions, transforming sequences of visual data into a structured narrative. This integrated approach bridges the gap between low-level visual perception and high-level semantic interpretation, enabling applications in video summarization, assistive technologies, and multimedia content generation. Experimental results on a benchmark video dataset demonstrate the potential of the system to produce meaningful and fluent storylines that reflect the visual content.

Keywords: Keyframe Detection · Object Detection · Scene Description · Video Summarization.

1 Introduction

Automatic scene understanding in video represents one of the most complex and relevant challenges at the intersection of computer vision and natural language processing. As the amount of audiovisual content grows exponentially, it becomes imperative to develop systems capable of interpreting, summarizing, and narrating visual events in an efficient and human-understandable manner. However, many current solutions present significant limitations: they require high computational resources, lack modularity, and do not offer sufficient interpretability for real-time applications or on devices with limited capabilities.

In this context, there is a need to develop a system that combines computer vision techniques and language models in an efficient, interpretable, and scalable manner. Automatic analysis of audiovisual content has gained increasing importance in multiple domains, from intelligent surveillance to robotic assistance and multimedia content generation. However, most current approaches to

video scene understanding suffer from critical shortcomings. For example, processing is performed frame by frame, resulting in high computational costs, they rely on monolithic architectures that are difficult to interpret, and they are not optimized for devices with limited computational resources. These shortcomings hinder their adoption in applications that demand efficiency and modularity. Furthermore, the absence of effective keyframe selection mechanisms leads to redundancy in visual processing, while the generation of semantic descriptions often lack coherent narrative structure and contextual relevance. The integration between visual perception and linguistic reasoning therefore remains a technical and conceptual challenge.

Given this scenario, we propose a multi-stage artificial intelligence pipeline that addresses these challenges through a modular architecture. The system begins with keyframe detection to reduce temporal redundancy, then object detection is performed using a lightweight detection model (YOLOv8n), and culminates with the generation of narrative descriptions using a large-scale language model (GPT-4o-mini), all integrated through the LangChain framework. This proposal seeks to bridge the gap between low-level visual perception and high-level semantic interpretation, enabling practical applications in domains such as robotics, intelligent surveillance, and augmented reality.

The main contributions of this work are the following:

- The design of a robust keyframe extraction method based on background subtraction and polynomial fitting.
- The implementation of an object detection system optimized for computationally limited environments.
- The integration of language models to generate structured narratives from visual data.

The remainder of the article is organized as follows: Section 2 reviews the most relevant related works; Section 3 presents the theoretical framework that supports each component of the system; Section 4 describes the proposed methodology in detail; Section 5 presents the experimental design and parameters used; Section 6 discusses the results obtained and their implications; and finally, Section 7 presents conclusions and possible future lines of research.

2 Related Work (Summary)

Recent progress in scene understanding has been driven by the integration of computer vision and natural language processing, enabling semantic interpretation over visual data. Object detection remains a foundational component in these systems. Jiménez-Bravo et al. [7] provide a comprehensive review of multi-object tracking (MOT) in vehicular traffic environments, noting that most approaches rely on resource-intensive end-to-end video processing, which limits their applicability in real-time, low-computation scenarios.

In parallel, the use of large language models (LLMs) in vision-language tasks has gained traction. Wang et al. [12] explore the robustness of machine learning

systems, showing that LLMs can enhance contextual resilience when integrated with visual data, albeit with increased computational costs. Similarly, Martinez-Martin and Del Pobil [8] emphasize the challenge of combining visual perception and semantic reasoning without compromising real-time performance.

Efforts to develop resource-efficient architectures have also emerged. Buenrostro-Mariscal and Santana-Mancilla [2] advocate for modular and domain-adaptive deep learning models, though they highlight a lack of practical multimodal implementations in mobile and embedded systems. Valladares-Poncela et al. [11] address this by optimizing LLM deployment in constrained industrial metaverse environments, demonstrating feasibility on mixed reality devices. Chatterjee et al. [4] introduce RoadSitu, a three-stage process for situation recognition in traffic videos using keyframe extraction to reduce redundancy. Wang et al. [13] propose VaVLM, a hybrid edge-cloud system that combines object detection with vision-language models to enable efficient video analysis on edge devices. Jha et al. [6] contribute with a Temporal Scene Understanding method that tracks objects across frames and generates interpretable scene narratives.

Despite these advances, several gaps remain. Many vision-language pipelines are too computationally demanding for mobile or edge deployment. Keyframe selection is rarely used as a preprocessing step, leading to inefficiencies in scene processing. Additionally, narrative generation is often monolithic and opaque, limiting modularity and interpretability [7] [12] [8]. To address these issues, we propose a modular, resource-efficient AI pipeline that extracts keyframes to reduce redundancy, applies object detection to identify relevant entities, and uses an LLM to generate coherent and structured scene narratives. By decoupling visual and semantic stages and focusing only on relevant frames, the system reduces computational cost and enhances explainability, making it suitable for real-time, low-cost applications in robotics, surveillance, and augmented reality.

3 Theoretical Framework

3.1 Keyframes Extraction

Content-based video indexing and retrieval has its foundations in the analysis of prime video temporal structures. Consequently, technologies for video segmentation and key-frame extraction have become crucial for the development of advanced digital video systems. Conventional algorithms for video partitioning and key-frame extraction are mainly implemented autonomously[3].

Background subtraction

One of the most effective strategies for keyframe extraction is based on motion analysis and change detection between consecutive frames. Background subtraction techniques are essential tools for this purpose, allowing the isolation of regions that experience substantial variations relative to an adaptive background model.

In this context, the pixel-wise Adaptive Gaussian Mixture Model (GMM), developed by Zivkovic and van der Heijden [16], represents a significant advance. This model estimates the color distribution of each pixel as a combination of Gaussians that are updated online, adapting to variations in lighting and changes in the background. The improvement introduced by Zivkovic [15], which includes a mechanism to automatically select the number of components of the mixture, increases the robustness and precision of the method in complex and dynamic scenarios.

In this work, we use OpenCV's MOG2 implementation [9], based on this GMM model, to generate a signal that quantifies the change in each frame. This signal is the basis for various keyframe selection algorithms that seek to identify local peaks, cumulative changes, and atypical clusters in the sequence, thus reflecting key moments within the video. Correctly identifying these keyframes is essential for achieving a faithful and efficient summary of visual content, facilitating downstream applications such as semantic search, indexing, and automatic scene analysis.

Polynomial Fitting

One of these strategies is polynomial fitting, which consists of approximating a continuous signal with a polynomial of degree d using least-squares regression techniques. This approach not only smooths the signal, but also allows one to obtain its analytical derivatives, which facilitates the identification of critical points (maxima, minima, and inflection points) that can be interpreted as candidates for key frames.

The theoretical basis of this methodology lies in the work of Gan, Ruan, and Mo [5], who developed a technique for baseline correction in chemical spectra. Their proposal, called *Iterative Polynomial Fitting with Automatic Threshold*, consists of iteratively fitting a polynomial curve to a signal and applying a dynamic threshold to separate the useful signal from background noise. This strategy has proven effective in preserving the signal's relevant characteristics while eliminating non-significant variations.

In the context of video analysis, a natural adaptation of this method is to apply the polynomial fitting to a previously obtained scoring signal (previously taken using MOG2) and then calculate the derivative of the fitted polynomial. The real roots of this derivative within the valid time interval correspond to local change points in the signal, which can be associated with relevant events in the sequence.

3.2 Object Detection

Object detection is a fundamental task in computer vision that combines object classification and localization within an image. Its main objective is to predict the classes of objects present and their locations using bounding boxes, and it is widely applied in domains such as autonomous vehicles, surveillance, and medical imaging. A key metric for evaluating detection performance is the Intersection

over Union (IoU), which measures the overlap between the predicted bounding box B and the ground truth box B_{gt}:

$$\text{IoU} = \frac{|B \cap B_{\text{gt}}|}{|B \cup B_{\text{gt}}|}$$

This metric is used during both training and evaluation to assess spatial accuracy. Additionally, the confidence score C quantifies the reliability of a detection by combining the probability of an object being present with the quality of the bounding box prediction [14]:

$$C = P(\text{object}) \cdot \text{IoU}(B, B_{\text{gt}})$$

A widely adopted approach for real-time object detection is the YOLO (You Only Look Once) algorithm, originally proposed by Redmon *et al.* in 2016. YOLO processes an input image of dimensions $A \times B \times C$ through a convolutional neural network (DarkNet) to extract features, which are then mapped into a grid and passed through fully connected layers to produce an output of dimension:

$$7 \times 7 \times (5B + C)$$

where B is the number of predicted bounding boxes per cell and C is the number of possible classes. Each cell in the 7×7 grid generates B predictions, each including bounding box coordinates (x, y, w, h), an object score (probability of object presence multiplied by IoU), and a class probability distribution [10].

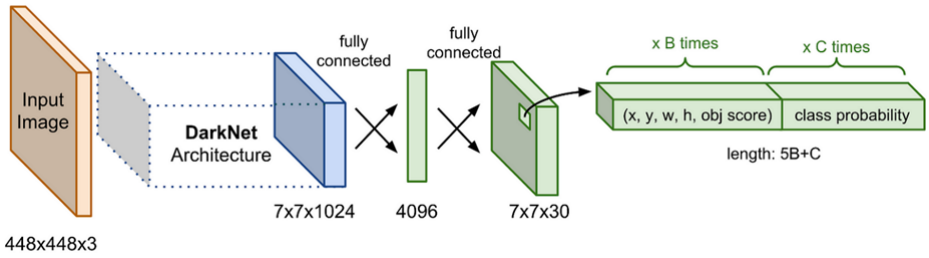


Fig. 1: Basic YOLO model architecture diagram [10].

4 Metodology

4.1 Video Processing

To prepare video frames for keyframe extraction, a series of preprocessing steps were applied. First, each frame was converted to grayscale using OpenCV's `cv2.COLOR_BGR2GRAY` transformation, which removes the alpha channel and applies the luminance formula $Y = 0.29\,R + 0.587\,G + 0.114\,B$. Then, frames were

resized to a resolution of 96×54 pixels using `cv2.resize()` to reduce computational cost while preserving the aspect ratio [1]. To improve stability in background modeling, Gaussian blur was applied using OpenCV's `GaussianBlur`, which attenuates high-frequency noise and enhances consistency between adjacent frames. For motion analysis, the MOG2 background subtraction algorithm [9] was used to generate a binary motion mask $M_t \in \{0, 255\}^{H \times W}$, where pixels with value 255 indicate significant changes. From this mask, a scalar activity score S_t was computed as:

$$S_t = \frac{1}{H \cdot W} \sum_{x=1}^{W} \sum_{y=1}^{H} \left(\frac{M_t(x, y)}{255} \right)$$

This temporal signal reflects the level of visual activity at each time step t and serves as input for keyframe selection algorithms, which identify relevant moments based on local peaks, clusters, or cumulative change patterns.

4.2 Keyframe Extraction

To detect significant transitions in the temporal scoring signal, a polynomial fitting method was applied, adapted from the baseline correction technique by Gan et al. [5]. The process begins by discarding low-activity values (below 0.1) to avoid noise influence. A polynomial $p(t)$ of degree d is fitted using least squares, and its derivative $p'(t)$ is computed. The real roots of $p'(t)$ within the valid time range are interpreted as candidate keyframes. To select the optimal degree, values $d \in [3, 20]$ are evaluated based on the root mean square error (RMSE) and a penalty applied when the number of detected keyframes falls outside the desired range $[n_{\min}, n_{\max}]$. The final score is computed as:

$$\text{Score} = \alpha \cdot \text{RMSE} + \beta \cdot \text{Penalty}$$

with $\alpha = 1.0$ and $\beta = 10.0$. This approach enables smooth modeling of temporal variation and robust identification of key moments in video sequences.

4.3 Object Detection

Object detection was performed using the YOLOv8 nano model (`yolov8n.pt`), a lightweight architecture optimized for real-time inference. The process was automated via a Python script using the Ultralytics library on a Windows 11 system with optional GPU support. The pipeline involved loading the model, iterating over input images, and applying the `model.predict(...)` method to extract object class, confidence score, and bounding box coordinates (x_1, y_1, x_2, y_2). For each image, the Intersection over Union (IoU) was computed between all bounding box pairs, and overlaps exceeding a threshold of 0.1 were recorded. The results were saved in a structured JSON file containing the list of detected objects and overlapping pairs per image. Additionally, annotated images with bounding boxes were automatically saved for visual verification. This JSON output served as input for the subsequent semantic interpretation stage.

4.4 Semantic Video Summarization

The semantic interpretation of visual content was performed using a LLM model from OpenAI, specifically `gpt-4o-mini`, integrated through the LangChain framework in Python. The input for this stage was a consolidated JSON file generated by the YOLOv8 object detection pipeline, containing detection results and calculated object collisions per image.

Workflow

1. **JSON Input Loading.** A single JSON file was loaded. This file contained detection and collision data structured by video folder and image name, as previously processed by the YOLOv8n model.
2. **Language Model Initialization.** The `gpt-4o-mini` model was initialized using `langchain_openai`, with the temperature parameter set to 0.0 to ensure deterministic outputs. The API key was loaded securely from environment variables using the `dotenv` library.
3. **Automated Interpretation Process.** A custom Python class was implemented to process each image's detection and collision data. For each image, a dynamic prompt in natural language was generated and submitted to the model. The model was instructed to return a plain JSON object with the following fields:
   - `general_description`: a brief summary of the detected object classes.
   - `objects_list`: a bulleted list with object counts by class.
   - `collision_analysis`: interpretation of overlaps based on Intersection over Union (IoU).
   - `narrative`: a narrative description of the scene in natural language.
4. **Prompting Strategy and Output Handling.** Prompts were carefully crafted to elicit valid JSON responses, with strict instructions to avoid additional formatting (e.g., Markdown or code blocks). If a response could not be parsed as valid JSON, the entire raw output was stored in the `narrative` field for further review, while other fields were left blank.
5. **Database Integration.** Each result was stored in a MySQL database in a table named `stories`, with columns corresponding to the JSON keys. This allowed for structured storage and later querying or visualization.

5 Experimental Design

5.1 Dataset Description

For the experimental validation of the proposed system, a set of 35 videos captured specifically for this study was used. The recordings were made by our team using the front camera of a Moto G54 smartphone, which features a 16-megapixel sensor and an aperture of f/2.4. All videos have a 1080p resolution at 30 fps, providing sufficient quality for visual analysis tasks under real-world conditions.

Each video is 30 seconds long and was recorded inside a supermarket, with the purpose of simulating a real environment where the keyframe detection algorithm is expected to be applied. This environment was chosen because of its visual complexity: It features multiple objects, constant movement of people, variability in textures, and dynamic lighting conditions, making it a demanding test case for video processing methods.

5.2 Parameters

Preprocessing and Scoring: We use OpenCV 4.x to preprocess video frames at a resolution of 96×54 pixels. Frames are converted to grayscale and smoothed with a Gaussian blur (kernel size 5×5). Scoring is performed via MOG2 background subtraction, and the score is computed as the normalized ratio of active pixels (value = 255) within the range [0,1].

Keyframe Extraction: Keyframes are identified by fitting a polynomial (degree 3–20) to the scoring signal and selecting the real roots of its first derivative. Optimization is guided by RMSE, with a penalty applied if the number of keyframes falls outside the range [2,30]. The hyperparameters used are $\alpha = 1.0$ and $\beta = 10.0$.

Object Detection (YOLOv8n): We employ the `yolov8n.pt` model from Ultralytics to detect objects in extracted keyframes (in .jpg or .png format). The output includes class labels, confidence scores, and bounding boxes in (xyxy) format. A collision is recorded if the Intersection over Union (IoU) exceeds 0.1.

Narrative Generation (LLM): Scene descriptions are generated using the `gpt-4o-mini` model via LangChain with a temperature of 0.0 for deterministic output. The input is a JSON file containing object detections and collision data, and the output is a structured JSON with fields for `general_description`, `objects_list`, `collision_analysis`, and `narrative`.

5.3 Experiments

To validate the system and fine-tune its parameters, a total of 10 videos, each approximately 30 seconds long, were processed through the following pipeline:

- **Preprocessing:** Visual enhancement using a custom filter algorithm.
- **Keyframe extraction:** Keyframes selected using the processor and saved as `.jpg`.
- **Object detection:** YOLOv8n model applied to each image, producing a JSON file per video folder containing detections and collisions.
- **Semantic interpretation:** Each JSON was processed by the GPT-4o-mini chatbot to generate structured summaries and narrative descriptions.

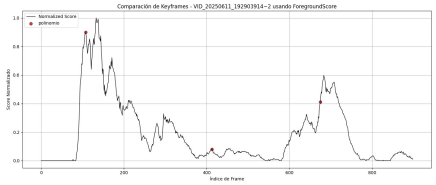
In total, 46 keyframes were analyzed, and corresponding automatic descriptions were generated. This phase confirmed the system's integrity, allowed adjustment of the detection model, and validated the coherence of the language model's outputs.

6 Results and Discussion

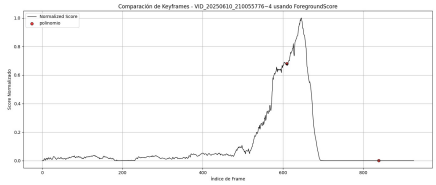
6.1 Keyframe Extraction

As observed in Figure 2, MOG2 provides an accurate representation of changes in videos with a static background, and the polynomial method yields good results in selecting frames that can be interpreted as representative.

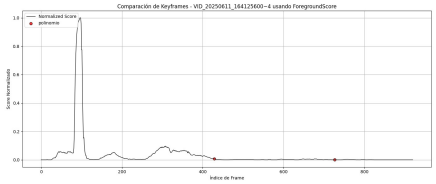
In the conducted experiments, both correct and incorrect selections of keyframes were observed. In the successful cases, the system accurately identified representative moments: in one instance, the first frame was selected at an initial peak, the second in a valley, and the third at a subsequent peak; in another, the first frame was chosen during the ascent to a summit and the second in a valley, effectively capturing meaningful transitions in the signal. Conversely, in the failed cases, the system selected two consecutive frames located in a valley before a significant peak, or multiple frames in valleys, with only the second and sixth frames aligning with peaks, resulting in missed key events. These examples highlight certain limitations of the method under specific temporal variability conditions.



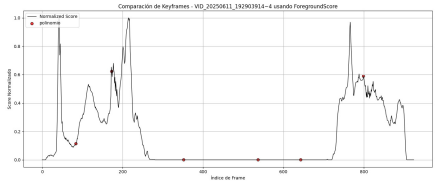
(a) Successful example 1 (`video/VID_20250611_192903914~2`)



(b) Successful example 2 (`video/VID_20250611_191323468~3`)



(c) Failed example 1 (`video/VID_20250611_164125600~4`)



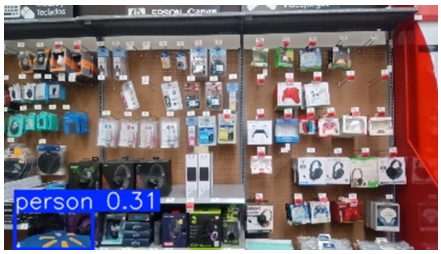
(d) Failed example 2 (`video/VID_20250611_192903914~4`)

Fig. 2: Keyframes selected by the polynomial method in a background subtraction model: detection results showing successful and failed examples.

6.2 Object Detection

As observed in Figure 3, the YOLOv8 nano model demonstrated reasonably accurate object detection on the keyframes extracted from the videos. Out of a

total of 46 keyframes, it correctly detected objects in 33 images (approximately 72%), mainly performing well on common classes such as people. However, the model struggled with less common objects (e.g., products), resulting in incorrect detections in 13 images (about 28%).



(a) Successful example 1 (frame_00051.jpg): correct detection of objects at a complicated angle.

(b) Successful example 2 (frame_00164.jpg): objects detected in the distance and with collisions.

(c) Failed example 1 (frame_00115.jpg): Incorrect object detection.

(d) Failed example 2 (frame_00184.jpg): Incorrect object detections at a complicated angle.

Fig. 3: YOLOV8N model detection results: Successful and failed examples.

6.3 Semantic Video Summarization

The narratives shown in Table 1 were generated based on object detection data provided in the prompt and JSON. Since the language model operates solely on structured data and does not access raw image content, it creates imaginative descriptions that sometimes do not fully match the visual content, highlighting the limitations of relying solely on structured data for scene interpretation.

7 Conclusions

This article presents a modular and efficient pipeline for semantic understanding of video scenes, integrating computer vision techniques and large-scale language

Table 1: Narratives generated from YOLO detections

Image	Objects List	Narrative
frame_00164.jpg	- People: 4	The image shows several people in a shopping center, possibly walking or waiting. The collision detected indicates that two of them are in a small space, which could reflect the typical congestion in areas of high influx such as corridors or waiting areas.
frame_00115.jpg	- People: 1 - Bottles: 1	In this image, we observe a person who seems to be enjoying their time in the mall, possibly exploring stores. The presence of a bottle suggests that it could be taking a break or moisturizing while making their purchases.
frame_00051.jpg	- People: 1	The image shows a person in a shopping center, possibly exploring stores or waiting for someone. The absence of collisions indicates that the flow of people is fluid, which contributes to a relaxed and pleasant atmosphere for buyers.
frame_00184.jpg	- People: 1 - Microwave: 1	In this image, we observe a person near a microwave, possibly in a rest area or food area of the mall. The presence of the microwave indicates that there could be a space for food preparation, while the person suggests that the place is being used, which adds life to the scene.

models. Keyframe extraction using motion analysis and polynomial fitting significantly reduced temporal redundancy, optimizing subsequent processing. Object detection with the YOLOv8n model demonstrated acceptable performance under real-world conditions, achieving a 72% hit rate for key images, especially for common classes such as people. Finally, narrative generation using GPT-4o-mini allowed structured data to be transformed into coherent and contextualized descriptions, demonstrating the potential of language models to enrich visual interpretation.

The experimental results validate the viability of the system in dynamic and uncontrolled environments, such as supermarkets, and highlight its applicability in low-computational scenarios, such as mobile devices or embedded systems. The proposed architecture not only improves processing efficiency but also enhances system interpretability and scalability.

Future work includes incorporating real-time feedback mechanisms to dynamically adjust keyframe selection, as well as improving the detection model to address less frequent classes or those with high visual variability.

References

1. Amanatiadis, A., Andreadis, I.: Performance evaluation techniques for image scaling algorithms. In: 2008 IEEE International Workshop on Imaging Systems and Techniques. pp. 114–118. IEEE, Chania, Greece (2008). https://doi.org/10.1109/IST.2008.4659952
2. Buenrostro-Mariscal, R., Santana-Mancilla, P.C., Montesinos-López, O.A., Nieto Hipólito, J.I., Anido-Rifón, L.E.: A review of deep learning applications for the next generation of cognitive networks. Applied Sciences **12**(12) (2022). https://doi.org/10.3390/app12126262, https://www.mdpi.com/2076-3417/12/12/6262
3. Calic, J., Izuierdo, E.: Efficient key-frame extraction and video analysis. In: Proceedings. International Conference on Information Technology: Coding and Computing. pp. 28–33. IEEE (2002). https://doi.org/10.1109/ITCC.2002.1000355

4. Chatterjee, S., Shin, H., Gil, J.M., Byun, Y.C.: Roadsitu: Leveraging road video frame extraction and three-stage transformers for situation recognition. Results in Engineering **24**, 103197 (2024). https://doi.org/https://doi.org/10.1016/j.rineng.2024.103197, https://www.sciencedirect.com/science/article/pii/S259012302401452X
5. Gan, F., Ruan, G., Mo, J.: Baseline correction by improved iterative polynomial fitting with automatic threshold. Chemometrics and Intelligent Laboratory Systems **82**(1-2), 59–65 (2006)
6. Jha, S.S., Garcia, K., Antille, Y.S., Solèr, M.E., Padua, S., Mayer, S.: Temporal scene understanding using contextually unique identification. In: 2024 IEEE 36th International Conference on Tools with Artificial Intelligence (ICTAI). pp. 1016–1023 (2024). https://doi.org/10.1109/ICTAI62512.2024.00145
7. Jiménez-Bravo, D.M., Álvaro Lozano Murciego, Sales Mendes, A., Sánchez San Blás, H., Bajo, J.: Multi-object tracking in traffic environments: A systematic literature review. Neurocomputing **494**, 43–55 (2022). https://doi.org/https://doi.org/10.1016/j.neucom.2022.04.087, https://www.sciencedirect.com/science/article/pii/S0925231222004672
8. Martinez-Martin, E., Pobil, A.P.D.: Robot vision for manipulation: A trip to real-world applications. IEEE Access **9**, 3471–3481 (2021). https://doi.org/10.1109/ACCESS.2020.3048053
9. OpenCV: Opencv: cv::backgroundsubtractormog2 class reference. https://docs.opencv.org/3.4/d7/d7b/classcv_1_1BackgroundSubtractorMOG2.html (2025), generated on Tue Jun 10 2025 by Doxygen 1.8.13. Accessed: 2025-06-10
10. Ragab, M.G., Abdulkadir, S.J., Muneer, A., Alqushaibi, A., Sumiea, E.H., Qureshi, R., Al-Selwi, S.M., Alhussian, H.: A comprehensive systematic review of yolo for medical object detection (2018 to 2023). IEEE Access **12**, 57815–57836 (2024)
11. Valladares-Poncela, A., Fraga-Lamas, P., Fernández-Caramés, T.M.: On-device automatic speech recognition for low-resource languages in mixed reality industrial metaverse applications: Practical guidelines and evaluation of a shipbuilding application in galician. IEEE Access **13**, 77017–77038 (2025). https://doi.org/10.1109/ACCESS.2025.3564137
12. Wang, Y., Sun, T., Li, S., Yuan, X., Ni, W., Hossain, E., Vincent Poor, H.: Adversarial attacks and defenses in machine learning-empowered communication systems and networks: A contemporary survey. IEEE Communications Surveys Tutorials **25**(4), 2245–2298 (2023). https://doi.org/10.1109/COMST.2023.3319492
13. Zhang, Y., Wang, H., Bai, Q., Liang, H., Zhu, P., Muntean, G.M., Li, Q.: Vavlm: Toward efficient edge-cloud video analytics with vision-language models. IEEE Transactions on Broadcasting pp. 1–13 (2025). https://doi.org/10.1109/TBC.2025.3549983
14. Zhang, Z., Lu, X., Cao, G., Yang, Y., Jiao, L., Liu, F.: Vit-yolo: Transformer-based yolo for object detection. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 2799–2808 (2021)
15. Zivkovic, Z.: Improved adaptive gaussian mixture model for background subtraction. In: Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on. vol. 2, pp. 28–31. IEEE (2004)
16. Zivkovic, Z., van der Heijden, F.: Efficient adaptive density estimation per image pixel for the task of background subtraction. Pattern Recognition Letters **27**(7), 773–780 (2006)