Simposio CEA de Robótica, Bioingeniería, Visión Artificial y Automática Marina 2025

RBVM
Simposio CEA de Robótica, Bioingeniería, Visión por Computador y Automática Marina

Segmentación semántica de instancias de objetos empleando un vocabulario abierto

Decena-Gimenez, M.*, Moncada-Ramirez, J., Ruiz-Sarmiento, J.R., Gonzalez-Jimenez, J.

Grupo de Percepción Artificial y Robótica Inteligente (MAPIR), Dept. de Ingeniería de Sistemas y Automática, Instituto Universitario en Ingeniería Mecatrónica y Sistemas Ciberfísicos (IMECH.UMA), Universidad de Málaga, Blvr. Louis Pasteur, 35, 29071 Málaga, España.

Resumen

La segmentación semántica de instancias tradicional, basada en modelos como Detectron2, está restringida por un "vocabulario cerrado" derivado de sus datos de entrenamiento (p. ej. COCO), lo que limita su capacidad para reconocer objetos de categorías no consideradas. Para superar esta limitación, presentamos TALOS, un método modular y flexible para la segmentación semántica de instancias con vocabulario abierto. TALOS ejecuta una secuencia de tres etapas: *Tagging* (extracción de etiquetas semánticas de las clases de objetos presentes), *Location* (localización de *bounding boxes* para cada instancia mediante *visual grounding* basado en las etiquetas) y *Segmentation* (generación de máscaras de píxeles precisas de forma agnóstica a la categoría). La modularidad permite integrar diversas tecnologías de vanguardia. Evaluaciones cualitativas demuestran que TALOS identifica y segmenta correctamente objetos de categorías ajenas a COCO, superando a Detectron2 en riqueza semántica y calidad de las máscaras, especialmente en escenas complejas.

Palabras clave: Visión por Computador, Reconocimiento de Objetos, Aprendizaje Profundo, Segmentación de Imágenes.

Instance semantic segmentation using an open vocabulary

Abstract

Traditional instance semantic segmentation, based on frameworks like Detectron2, is restricted by a "closed vocabulary" derived from its training data (e.g., COCO), limiting its ability to recognize objects from unseen categories. To overcome this limitation, we present TALOS, a modular and flexible method for open-vocabulary instance semantic segmentation. TALOS executes a sequence of three stages: Tagging (extraction of semantic labels of present object classes), Location (bounding box localization for each instance via visual grounding based on the extracted labels), and Segmentation (generation of accurate pixel masks in a category-agnostic manner). Modularity allows integrating diverse state-of-the-art technologies. Qualitative evaluations demonstrate that TALOS correctly identifies and segments objects from categories beyond COCO, outperforming Detectron2 in semantic richness and mask quality, especially in complex scenes.

Keywords: Computer Vision, Object recognition, Deep Learning, Image Segmentation.

1. Introducción

La percepción precisa del entorno mediante Visión por Computador es crucial para sistemas autónomos complejos, como vehículos autónomos o robots móviles asistenciales (Luperto et al., 2023, 2019). Dentro del análisis de escenas visuales, dos tareas fundamentales, aunque distintas, son la detección de objetos y la segmentación semántica de instancias. La primera se enfoca en localizar objetos mediante cajas contenedoras (*bounding boxes*) y clasificarlos según categorías predefinidas. La segunda, más detallada, busca no solo clasificar, sino también delinear la forma exacta a nivel de píxel cada instancia individual de objeto presente en la imagen.

Actualmente, los métodos más exitosos para estas tareas, como Mask-RCNN (He et al., 2017) o Detectron2 (Wu et al., 2019), se basan en modelos de Aprendizaje Profundo (*Deep Learning*, DL). Estos modelos requieren ser entrenados con grandes repositorios de imágenes anotadas, conocidos como conjuntos de entrenamiento. Sin embargo, esta dependencia de los datos de entrenamiento impone una limitación significativa: los modelos aprenden a detectar exclusivamente las categorías

*Autor para correspondencia: macorisd@uma.es

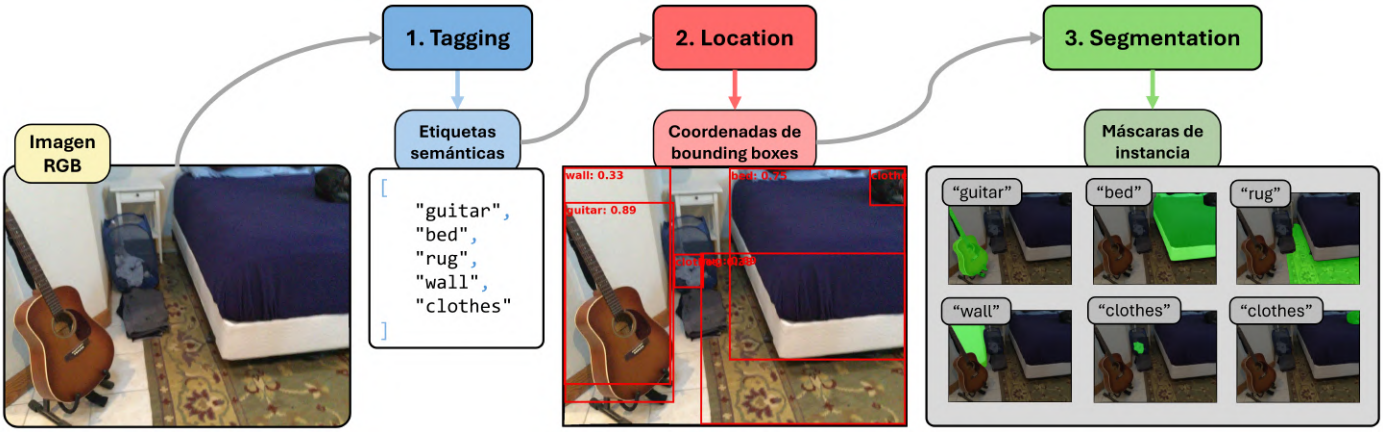

Figura 1: Diagrama general del método de segmentación semántica de instancias de objetos.

presentes en dichos datos, operando con lo que se conoce como un "vocabulario cerrado". Son incapaces de reconocer objetos pertenecientes a categorías no vistas durante el entrenamiento. Conjuntos de datos populares como COCO (Lin et al., 2014) (con 80 categorías cotidianas) o Cityscapes (Cordts et al., 2016) (con 30 clases de entornos urbanos) son ampliamente utilizados. Un sistema entrenado exclusivamente con COCO, por ejemplo, puede identificar categorías predefinidas como plátanos, naranjas o manzanas, pero fallará al detectar cualquier otro tipo de fruta, o lo clasificará erróneamente. Si bien es posible mitigar esto mediante ajuste fino (*fine-tuning*) para incluir nuevas categorías, este proceso a menudo reduce el rendimiento general y genera soluciones ad-hoc difícilmente generalizables a nuevos contextos.

Frente a esta limitación, surge la necesidad de emplear sistemas con vocabulario abierto. Algunas herramientas existentes en esta línea requieren, además de la imagen, una lista explícita de las etiquetas semánticas a buscar. Otras más recientes, como TAG (Kawano and Aoki, 2024), operan directamente sobre la imagen, pero pueden ofrecer capacidades limitadas de personalización sin recurrir a un costoso reentrenamiento.

Para abordar las deficiencias del vocabulario cerrado y mejorar la flexibilidad de las soluciones de vocabulario abierto, este artículo presenta TALOS (TAgging-LOcation-Segmentation). Se trata de un método modular que realiza segmentación semántica de instancias con vocabulario abierto partiendo únicamente de una imagen RGB como entrada (véase la Figura 1). TALOS integra distintas tecnologías de vanguardia en tres etapas secuenciales:

- ***Tagging***: Extrae etiquetas semánticas de las categorías de objetos presentes en la imagen usando modelos a gran escala.
- ***Location***: Localiza los *bounding boxes* para las instancias detectadas de cada etiqueta generada, mediante técnicas de *visual grounding*.
- ***Segmentation***: Genera máscaras de píxeles precisas para cada instancia localizada, utilizando segmentación agnóstica a la categoría.

El diseño modular y flexible de TALOS permite integrar y actualizar fácilmente diferentes tecnologías para cada etapa. Este método, disponible públicamente en GitHub (Decena-Gimenez, 2025), ha sido evaluado cualitativamente. Los resultados demuestran su capacidad para generar segmentaciones válidas de instancias con vocabulario abierto, superando en riqueza semántica y fidelidad de contornos a herramientas populares basadas en vocabulario cerrado como Detectron2, resaltando así el potencial de los enfoques sin limitaciones de categorías semánticas predefinidas.

2. Método para segmentación semántica de instancias de objetos con vocabulario abierto

El método desarrollado, TALOS, toma como entrada una imagen RGB y produce como salida un conjunto de detecciones de instancias de objetos, compuestas por una máscara de segmentación y una categoría semántica (véase la Figura 1). Se divide en tres fases secuenciales. La primera fase, *Tagging*, procesa una imagen de entrada y produce una lista de las etiquetas semánticas de los objetos presentes en la imagen (véase la Sección 2.1). La segunda fase, *Location*, parte de la imagen de entrada y de la lista de etiquetas semánticas obtenidas en la etapa anterior y genera, para cada una de ellas, los *bounding boxes* correspondientes a las instancias de objeto presentes en la imagen (véase la Sección 2.2). La tercera fase, *Segmentation*, parte de la imagen de entrada y de los *bounding boxes* localizados en la etapa anterior y genera las máscaras de segmentación de cada instancia de objeto detectada (véase la Sección 2.3).

Al tratarse de un método modular, cada fase es independiente y está desacoplada de las demás, lo que permite implementar diferentes tecnologías del estado del arte en cada una de ellas.

2.1. *Tagging (etiquetado semántico)*

El objetivo de la etapa de *Tagging* es el de, partiendo de una imagen de entrada, producir una lista de etiquetas semánticas de las categorías de objetos presentes en dicha imagen. Formalmente, la función que modelaría esta etapa de *Tagging* se puede definir como:

$$\mathcal{T}(I) = L_T \tag{1}$$

La entrada de la función $\mathcal{T}(\cdot)$ (1) es una imagen $I \in \mathbb{R}^{H\times W\times 3}$, donde H y W representan la altura y la anchura respectivamente, y los tres canales corresponden a la codificación RGB.

La salida de la función $\mathcal{T}(\cdot)$ es un conjunto de etiquetas textuales $L_T = \{t_1, t_2, \ldots, t_n\}$, tal que $\forall i \in [1, n], t_i \in V$, donde V representa el vocabulario disponible. Este vocabulario V, aunque finito (el número de categorías de objetos posibles en el mundo real así lo es), es extremadamente amplio, lo que caracteriza al enfoque de vocabulario abierto. La cardinalidad de L_T depende del número de categorías de objetos presentes en la imagen de entrada. Programáticamente, esta salida L_T se puede almacenar en formato JSON como una lista de cadenas de caracteres. A continuación se proponen dos alternativas para la implementación de esta etapa.

2.1.1. *Primer método de Tagging: Etiquetado directo*

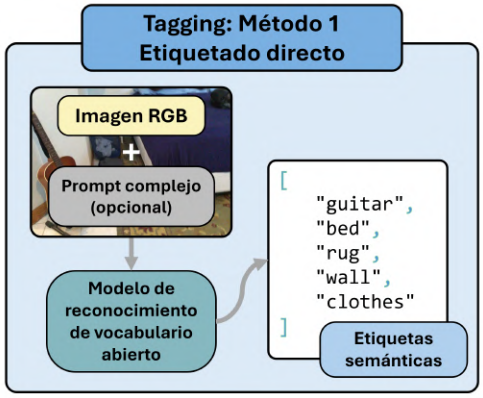

Figura 2: Diagrama de funcionamiento del método de *Tagging* basado en etiquetado directo.

El primer método propuesto para la etapa de *Tagging* es la extracción directa de etiquetas semánticas mediante un modelo capaz de producir esta salida ante una imagen de entrada (véase la Figura 2).

El modelo empleado para este fin puede ser uno como RAM++ (Huang et al., 2023), que precisa únicamente de una imagen de entrada para producir las etiquetas semánticas de los objetos presentes en la escena. No obstante, resulta especialmente provechoso realizar un etiquetado directo usando un Modelo de Visión-Lenguaje a Gran Escala (del inglés *Large Vision-Language Model*, LVLM), necesitándose para ello un *prompt* en lenguaje natural. Esto último es realmente interesante dado el amplio contexto que se le puede proporcionar al modelo sobre la escena a analizar, personalizando el análisis en función de la tarea a realizar. Por ejemplo, si se trabaja con entornos de interiores, es posible y sencillo proporcionarle esta información contextual al modelo, para que trabaje de forma más orientada al reconocimiento de objetos específicos de estos espacios. Otro ejemplo es la posibilidad de solicitar, de forma aproximada, el número de elementos de la lista L_T, es decir, indicar al modelo que describa, como mucho, las cinco categorías de objetos predominantes de la escena, si es eso lo que se busca.

El uso de un *prompt* complejo y detallado para el *Tagging* directo mediante un LVLM resulta especialmente adecuado en este contexto. En primer lugar, al ser un método automatizado, es necesario que la salida del LVLM siga un formato estructurado, como JSON, para procesar los resultados. En segundo lugar, es conveniente indicar al LVLM ciertas pautas generales que le ayuden a cumplir su tarea, como escribir las etiquetas en singular (p. ej., "zapato" es más adecuado que "par de zapatos" al trabajar con instancias de objetos), o evitar características demasiado específicas para mitigar disparidades entre las etiquetas referidas al mismo objeto en sucesivas detecciones, si las hubiera (p. ej., "sofá" es más apropiado que "gran sofá rojo").

El uso de un LVLM en un sistema de reconocimiento de vocabulario abierto es adecuado, pues es sabido que uno de los aspectos más provechosos de la obtención de información visual mediante LVLMs es que, al estar entrenados con cantidades masivas de datos con pares imagen-texto de múltiples fuentes multimedia, no están limitados a utilizar etiquetas semánticas fijas en sus descripciones, sino que usan un catálogo realmente amplio de información semántica.

Un concepto adicionalmente propuesto para aumentar la calidad de la salida de esta etapa cuando se emplea un LVLM es el de la auto-reflexión (Moncada-Ramirez et al., 2025). Este concepto sugiere realizar una o varias consultas adicionales al modelo de gran escala proporcionándole su propia primera respuesta y preguntándole sobre la corrección y calidad de dicha salida. De esta forma, el modelo es capaz de "reflexionar" sobre respuestas anteriores, devolviendo contestaciones optimizadas en caso de detectar posibles mejoras.

2.1.2. *Segundo método de Tagging: LVLM descriptor de imagen + LLM extractor de palabras clave*

El segundo método propuesto para la etapa de *Tagging* está basado en el aprovechamiento de LVLMs y Modelos de Lenguaje a Gran Escala (del inglés *Large Language Model*, LLM). En concreto, un LVLM genera una descripción en lenguaje natural de la imagen de entrada y, posteriormente, un LLM realiza una extracción de palabras clave (etiquetas semánticas) a partir de dicha descripción.

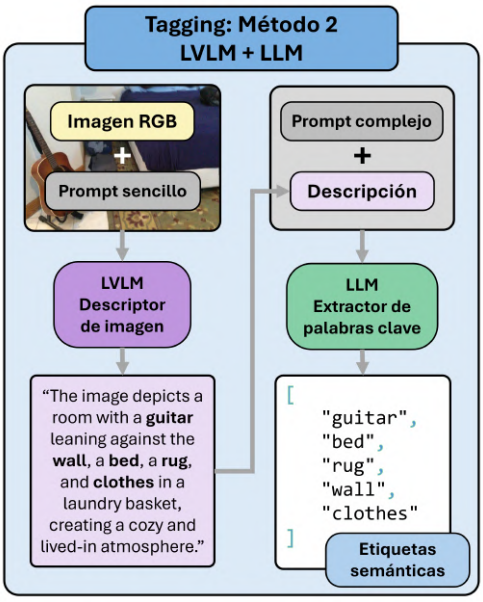

Figura 3: Diagrama de funcionamiento del método de *Tagging* basado en la combinación secuencial de modelos a gran escala.

Pese al alto éxito de los LVLMs del estado del arte en la tarea de descripción de imágenes, ciertos modelos multimodales a gran escala tienen dificultades a la hora de razonar sobre el

contexto e instrucciones del *prompt* de entrada, especialmente aquellos modelos con un número de parámetros más reducido (Ghosh et al., 2024). Ante esta situación, se plantea un método alternativo de *Tagging* que combina la ventaja del uso de un LVLM para la descripción textual de una imagen, mediante un prompt sencillo, con la buena capacidad de razonamiento de un LLM para la extracción de las palabras clave a partir de la descripción textual, utilizando un prompt más complejo, similar al propuesto en la Sección 2.1.1 (véase la Figura 3).

Este enfoque abre la posibilidad de emplear un modelo descriptor de imágenes más ligero (con menos parámetros). También ofrece la oportunidad de usar el mismo modelo multimodal para ambas fases, es decir, lograr que un mismo modelo describa la imagen y, posteriormente, extraiga las palabras clave de la descripción textual. Asimismo, el proceso de auto-reflexión mencionado en la Sección 2.1.1 es igualmente aplicable para el LLM extractor de palabras clave.

2.2. *Location (localización de instancias de objetos)*

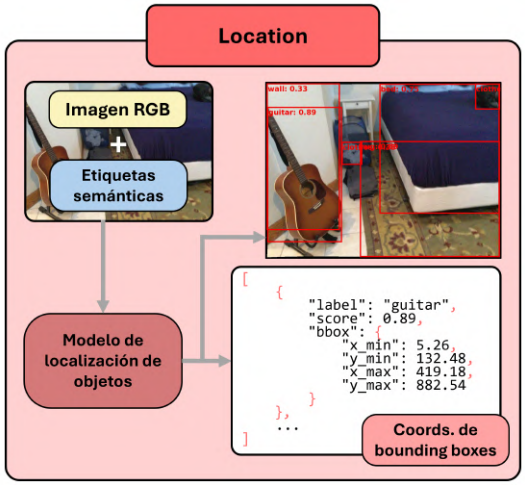

Figura 4: Diagrama de funcionamiento de la etapa de Location.

El objetivo de la etapa de *Location* es el de, partiendo de una imagen de entrada y de una lista de etiquetas semánticas de los objetos presentes en la imagen obtenida en la etapa de *Tagging*, producir una lista de instancias de objetos localizados en la imagen. Cada instancia contendrá una etiqueta semántica, las coordenadas donde se ha localizado y la confianza de la localización. Formalmente, la función que modelaría esta etapa de *Location* se puede definir como:

$$\mathcal{L}(I, L_T) = L_L \tag{2}$$

La entrada de la función $\mathcal{L}(\cdot)$ (2) está compuesta por:

- Una imagen $I \in \mathbb{R}^{H \times W \times 3}$.
- Un conjunto de etiquetas semánticas $L_T = \{t_1, t_2, \ldots, t_n\}$, obtenido como salida en la etapa de *Tagging*.

La salida de la función $\mathcal{L}(\cdot)$ es una lista de instancias localizadas de objetos $L_L = \{l_1, l_2, \ldots, l_n\}$, donde cada elemento $l_i \in L_L$ es una terna $(\text{label}_i, \text{score}_i, \text{bbox}_i)$, donde:

- $\text{label}_i \in L_T$ es la etiqueta semántica del objeto i.
- $\text{score}_i \in [0, 1]$ es el valor de confianza de la localización de label_i en bbox_i dentro de I.
- $\text{bbox}_i \in \mathbb{R}^4$ contiene las coordenadas del *bounding box* del objeto i en formato $(x_{\min}, y_{\min}, x_{\max}, y_{\max})$.

De forma programática, la salida L_L puede almacenarse en formato JSON, como una lista de diccionarios (véase la Figura 4).

El método propuesto para la etapa de *Location* emplea un modelo de *visual grounding*, que se refiere al proceso de establecer una conexión precisa entre descripciones textuales y regiones específicas de una imagen. Modelos de este tipo, como Grounding DINO (Liu et al., 2024), son capaces de, partiendo de una imagen de entrada y una lista de etiquetas semánticas de los objetos presentes en la imagen, localizar esos objetos, proveyendo además un valor de confianza en la localización. Resulta esencial, una vez más, que este modelo haya sido entrenado con un vasto repositorio de datos visuales y sus correspondencias textuales, para que se considere de vocabulario abierto.

2.3. *Segmentation (segmentación de instancias de objetos)*

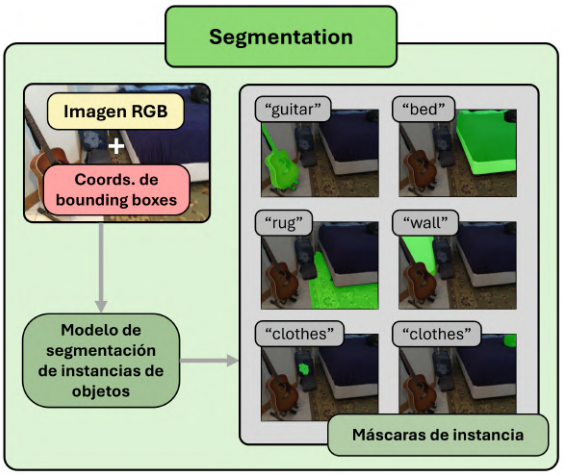

Figura 5: Diagrama de funcionamiento de la etapa de Segmentation.

La finalidad de la etapa de *Segmentation* es la de, partiendo de una imagen RGB de entrada y de las coordenadas de los *bounding boxes* de los objetos presentes en la imagen, proporcionadas por la etapa de *Location*, producir las máscaras binarias de segmentación de cada instancia de objeto localizada. Formalmente, la función que modelaría esta etapa resulta:

$$\mathcal{S}(I, L_L) = L_S \tag{3}$$

La entrada de la función $\mathcal{S}(\cdot)$ (3) está compuesta por:

- Una imagen $I \in \mathbb{R}^{H \times W \times 3}$.
- Una lista de n instancias localizadas de objetos L_L, obtenida como resultado de la etapa de *Location*.

La salida de la función $\mathcal{S}(\cdot)$ es una lista de máscaras de segmentación $L_S = \{m_1, m_2, \ldots, m_n\}$, donde cada $m_i \in L_S$ es una máscara binaria bidimensional de tamaño $H \times W$. En cada máscara m_i, los píxeles pertenecientes a la instancia $l_i \in L_L$ están marcados con valor 1, mientras que el resto de los píxeles toma el valor 0, lo que supone una segmentación precisa a nivel

de píxel para cada objeto. Cada máscara puede almacenarse de forma programática como un *array* bidimensional.

Cabe destacar que existe una correspondencia biunívoca entre L_L y L_S. Es decir, a cada instancia localizada de L_L le corresponde exactamente una máscara en L_S, lo que garantiza una asociación directa entre la localización de un objeto, que posee además información semántica, y su segmentación.

El método planteado para la etapa de *Segmentation* hace uso de un modelo especializado en segmentación de instancias de objetos en imágenes. Modelos de este tipo, como SAM2 (Ravi et al., 2024), son capaces de, partiendo de una imagen de entrada y las coordenadas de un *bounding box* en donde se estima que hay una instancia de objeto, producir una máscara binaria a nivel de píxel del objeto en cuestión, de forma agnóstica a su categoría semántica.

Así, el resultado para cada instancia i del proceso de segmentación semántica de instancias de objetos de TALOS es la máscara binaria m_i y su correspondiente label_i, junto con su *bounding box* localizado bbox_i y la confianza de la localización, score_i.

3. Detalles de implementación

TALOS ha sido desarrollado en Python y diseñado con un enfoque modular, donde cada etapa utiliza modelos específicos del estado del arte que pueden ser intercambiados o actualizados de forma independiente sin afectar al resto del sistema.

Para el método de *Tagging* con etiquetado directo se ha incorporado el LVLM Qwen 2.5 VL Instruct (Bai et al., 2025) y, adicionalmente, el modelo RAM++ (Huang et al., 2023). Para la etapa de *Tagging* con el método de LVLM + LLM se ha integrado LLaVA (Liu et al., 2023) como LVLM descriptor de imágenes y DeepSeek R1 (Guo et al., 2025) como LLM extractor de palabras clave. Para la etapa de *Location* se ha incluido la herramienta Grounding DINO (Liu et al., 2024). Por último, en la etapa de *Segmentation* se ha integrado el modelo SAM2 (Ravi et al., 2024).

A medida que se validen nuevos modelos en TALOS, se incorporarán al sistema y se publicarán en su repositorio de GitHub.

4. Experimentos

Para demostrar la efectividad del método propuesto para la segmentación semántica de instancias de objetos con vocabulario abierto, se ha realizado una validación cualitativa comparado sus resultados en distintos escenarios de interés con los del popular detector de vocabulario cerrado Detectron2, entrenado en base a las 80 categorías del popular dataset COCO. Para las siguientes pruebas del método TALOS se ha empleado:

- *Tagging* mediante LVLM + LLM: LLaVA (34b) como LVLM descriptor de imagen y DeepSeek R1 (14b) como LLM extractor de palabras clave.
- *Location*: Grounding DINO base.
- *Segmentation*: SAM2 Hiera Large.

A continuación se muestran tres ejemplos interesantes del rendimiento de ambos sistemas ante las mismas imágenes de entrada. Téngase en cuenta que los resultados de TALOS, al usar modelos a gran escala, no son deterministas; esto es, pueden variar entre ejecuciones, aunque las diferencias no suelen ser demasiadas (depende, también, de las tecnologías concretas que se empleen y los valores de sus parámetros).

4.1. *Primera prueba: Rendimiento ante categorías ajenas a COCO*

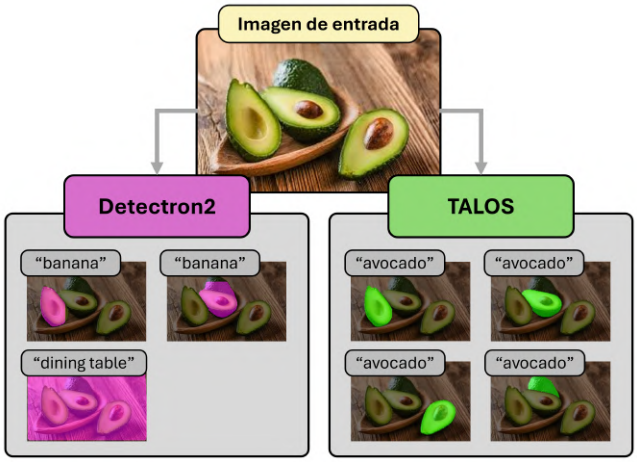

Figura 6: Comparación de la salida de Detectron2 y TALOS ante una imagen con una categoría ajena a COCO.

La categoría “*avocado*” (aguacate) no pertenece a COCO. Como se puede observar en la Figura 6, Detectron2 etiqueta algunos de los aguacates presentes en la imagen como plátanos y provee una máscara incorrecta para la posible mesa, mientras que TALOS detecta y segmenta correcta y exclusivamente los aguacates.

4.2. *Segunda prueba: Calidad de máscaras ante objetos con contornos complejos*

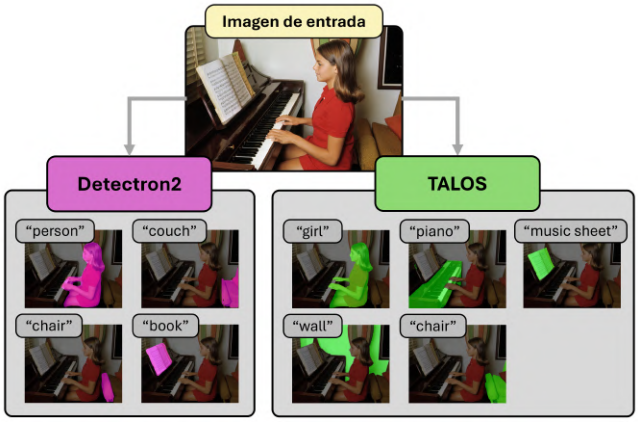

Figura 7: Comparación de la salida de Detectron2 y TALOS ante una imagen con objetos con contornos complejos.

La categoría “*piano*” no está presente en COCO, mas es detectada por TALOS, como se observa en la Figura 7. TALOS

también detecta de forma específica la partitura, a diferencia de Detectron2. Por otra parte, cabe destacar la diferencia cualitativa de las máscaras ante contornos complejos, como en el de la persona tocando el piano (véase la Figura 8).

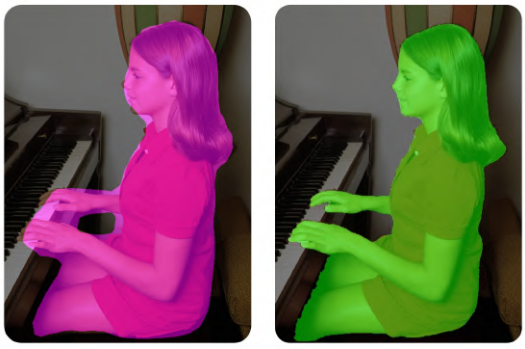

Figura 8: Comparación de las máscaras generadas por Detectron2 y TALOS ante un contorno complejo.

4.3. Tercera prueba: Rendimiento ante una multitud de objetos

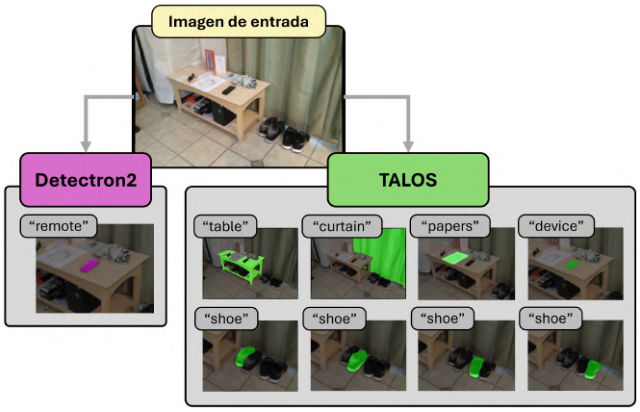

Figura 9: Comparación de la salida de Detectron2 y TALOS ante una imagen con multitud de objetos.

La imagen de la Figura 9 proviene del popular conjunto de datos ScanNet (Dai et al., 2017), donde se puede observar que Detectron2 presenta un rendimiento muy pobre. Únicamente ha logrado detectar un objeto, mientras que TALOS detecta y segmenta correctamente incluso los zapatos (complicados, dado su pequeño tamaño y su color negro), haciéndolo además de forma independiente para cada instancia en lugar de agruparlos.

5. Conclusiones

El trabajo presentado surgió del estudio de los sistemas de segmentación semántica de instancias de objetos, específicamente de sus limitaciones de etiquetas semánticas, y de la necesidad de construir sistemas que trabajen con un vocabulario abierto. Para ello, se ha propuesto TALOS, un método que considera un vocabulario abierto y ofrece la posibilidad de integrar tecnologías distintas en cada una de sus tres etapas (*Tagging*, *Location* y *Segmentation*) para lograr un mayor rendimiento, brindando además una posible personalización en lenguaje natural mediante el aprovechamiento de los modelos a gran escala. Los experimentos realizados han demostrado cualitativamente que TALOS es una herramienta que aporta mayor riqueza semántica y fidelidad en la generación de máscaras que herramientas populares de vocabulario cerrado del estado del arte.

Como trabajo futuro se plantea la integración del método TALOS en el sistema de reconocimiento de un robot móvil, de tal manera que este pueda identificar los elementos de su lugar de trabajo.

Agradecimientos

Este trabajo ha sido desarrollado en el contexto de los proyectos MINDMAPS (PID2023-148191NB-I00) y Voxeland (JA.B1-09), financiados por el Ministerio de Ciencia e Innovación y la Universidad de Málaga, respectivamente.

Referencias

Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., et al., 2025. Qwen2. 5-vl technical report. arXiv preprint arXiv:2502.13923.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B., 2016. The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3213–3223.

Dai, A., Chang, A. X., Savva, M., Halber, M., Funkhouser, T., Nießner, M., 2017. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5828–5839.

Decena-Gimenez, M., 2025. Talos: A modular and automatic system for open-vocabulary semantic instance segmentation. `https://github.com/macorisd/TALOS`.

Ghosh, S., Evuru, C. K. R., Kumar, S., Tyagi, U., Nieto, O., Jin, Z., Manocha, D., 2024. Visual description grounding reduces hallucinations and boosts reasoning in lvlms. arXiv preprint arXiv:2405.15683.

Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., et al., 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. arXiv preprint arXiv:2501.12948.

He, K., Gkioxari, G., Dollár, P., Girshick, R., 2017. Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969.

Huang, X., Huang, Y.-J., Zhang, Y., Tian, W., Feng, R., Zhang, Y., Xie, Y., Li, Y., Zhang, L., 2023. Open-set image tagging with multi-grained text supervision. arXiv preprint arXiv:2310.15200.

Kawano, Y., Aoki, Y., 2024. Tag: Guidance-free open-vocabulary semantic segmentation. IEEE Access.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C. L., 2014. Microsoft coco: Common objects in context. In: Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13. Springer, pp. 740–755.

Liu, H., Li, C., Wu, Q., Lee, Y. J., 2023. Visual instruction tuning. Advances in neural information processing systems 36, 34892–34916.

Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Jiang, Q., Li, C., Yang, J., Su, H., et al., 2024. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In: European Conference on Computer Vision. Springer, pp. 38–55.

Luperto, M., Monroy, J., Renoux, J., Lunardini, F., Basilico, N., Bulgheroni, M., Cangelosi, A., Cesari, M., Cid, M., Ianes, A., et al., 2023. Integrating social assistive robots, iot, virtual communities and smart objects to assist at-home independently living elders: the movecare project. International Journal of Social Robotics 15 (3), 517–545.

Luperto, M., Monroy, J., Ruiz-Sarmiento, J. R., Moreno, F.-A., Basilico, N., Gonzalez-Jimenez, J., Borghese, N. A., sep 2019. Towards long-term deployment of a mobile robot for at-home ambient assisted living of the elderly. In: European Conference on Mobile Robots (ECMR). DOI: `10.1109/ECMR.2019.8870924`

Moncada-Ramirez, J., Matez-Bandera, J.-L., Gonzalez-Jimenez, J., Ruiz-Sarmiento, J.-R., 2025. Agentic workflows for improving large language model reasoning in robotic object-centered planning. Robotics 14 (3), 24.

Ravi, N., Gabeur, V., Hu, Y.-T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., et al., 2024. Sam 2: Segment anything in images and videos. arXiv preprint arXiv:2408.00714.

Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., Girshick, R., 2019. Detectron2. `https://github.com/facebookresearch/detectron2`.