5. SAM: Segment Anything (v1, v2, v3)

Segmentación universal en imagen y vídeo, con prompts.

SAM (Segment Anything Model) es la evolución del modelo de segmentación más versátil: segmenta cualquier objeto en cualquier imagen mediante prompts simples (puntos, cajas, máscaras), y mantiene coherencia temporal en vídeo sin entrenar para clases específicas.

¿Qué es segmentación?

La segmentación de imágenes es una tarea básica en visión artificial que va más allá de la simple detección. Su objetivo es asignar una etiqueta de clase a cada píxel de una imagen, delimitando el contorno exacto de los objetos y el fondo.

Tipos de segmentación

Semántica

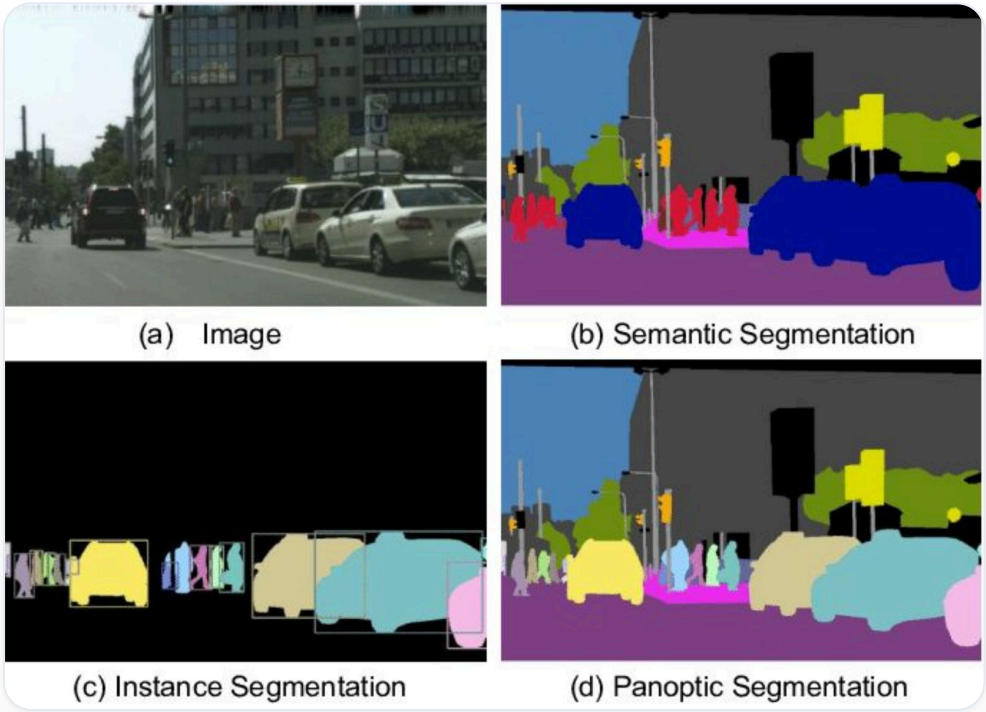
Cada píxel se clasifica en una categoría (por ejemplo, «coche», «persona», «fondo»). No distingue entre instancias individuales del mismo tipo de objeto: todos los coches reciben la misma máscara y etiqueta.

Por instancias

Cada objeto individual detectado recibe una máscara única y su propia etiqueta. Permite distinguir entre objetos del mismo tipo, como «coche 1», «coche 2», «persona A», «persona B».

Panóptica

Combina la segmentación semántica y por instancias. Asigna una máscara única y una etiqueta a cada instancia de objeto cubriendo toda la imagen. Es la más completa pero también la más compleja.



Arquitectura de SAM 1

Un diseño eficiente basado en tres componentes desacoplados.

Image encoder (el componente pesado)

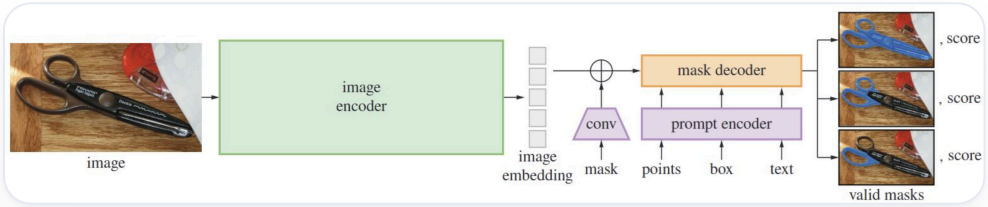
- Basado en Vision Transformer (ViT-H) preentrenado con MAE.
- Se ejecuta una única vez por imagen, generando un embedding reutilizable.
- Costoso computacionalmente, pero desacoplado de la interacción.

Prompt encoder (el componente ligero)

- Codifica entradas en tiempo real: puntos, cajas y texto (con positional encodings).
- También procesa máscaras previas (con convoluciones).
- Convierte la intención del usuario en embeddings compatibles.

Mask decoder (el cerebro rápido)

- Arquitectura Transformer ligera (two-way transformer).
- Realiza cross-attention entre embeddings de imagen y prompts.
- Predice 3 máscaras de salida (whole, part, subpart) para resolver ambigüedad.
- Latencia muy baja (unos 50 ms en navegador web).



SAM 1 admite puntos, cajas y máscaras de forma nativa. El texto como prompt requiere un modelo externo, por ejemplo Grounding DINO o CLIP, que traduzca la frase a una indicación espacial.

Arquitectura de SAM 2

Unificando imágenes y vídeo con memoria temporal.

Encoder jerárquico (Hiera): la evolución

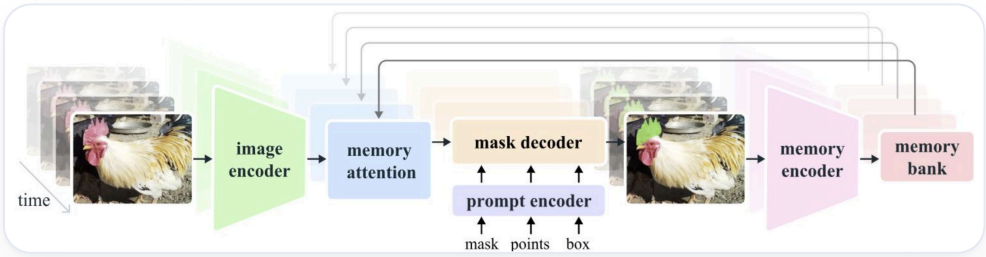
- Sustituye al ViT plano de SAM 1 por una arquitectura jerárquica (Hiera), un enfoque más potente.
- Procesa imágenes en múltiples escalas, capturando detalles finos y contexto global de forma simultánea.
- Mucho más eficiente, permitiendo el procesamiento de vídeo en tiempo real (streaming processing).
- Calcula los features del fotograma una sola vez, optimizando el rendimiento.

Mecanismo de memoria: la revolución

- Componente clave que permite entender el vídeo y mantener la coherencia temporal.
- **Memory bank:** almacena información de fotogramas pasados y del fotograma inicial (el prompt).
- **Memory attention:** permite al modelo «mirar atrás» para mantener la identidad del objeto a lo largo del tiempo.
- Maneja oclusiones, recordando el objeto incluso si desaparece momentáneamente de la vista.

Decodificador de máscaras y prompting: la interacción

- Recibe tanto los prompts del usuario (clics, cajas) como el contexto proporcionado por la memoria.
- Predice máscaras para el fotograma actual y actualiza la memoria para futuros procesamientos.
- **Ambiguity-aware:** genera 3 opciones de máscara si hay ambigüedad en la segmentación.
- Capaz de corregir la segmentación de todo el vídeo con una única corrección en un solo fotograma (propagación).



Arquitectura de SAM 3

Unificación de detección y tracking con desacoplamiento de reconocimiento.

Backbone unificado y prompting multimodal

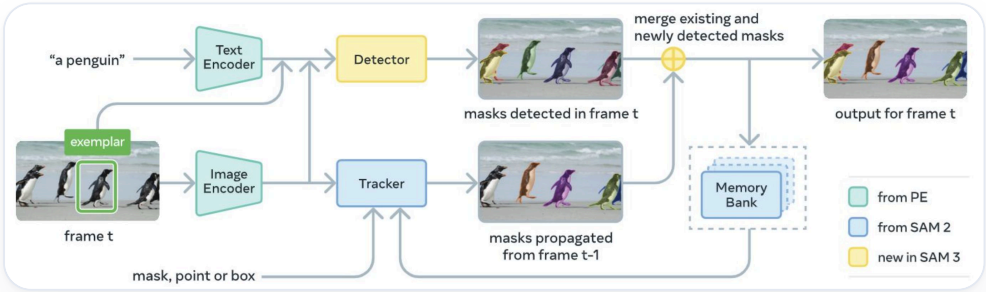
- Utiliza un Perception Encoder (PE) compartido, preentrenado con alineación visión-lenguaje (contrastive learning).
- **Nuevos prompts:** acepta texto (frases nominales como «un gato rayado») e image exemplars (ejemplos visuales positivos o negativos).
- **Fusion encoder:** un nuevo módulo que condiciona los embeddings de imagen con los tokens del prompt antes de la decodificación.

Diseño dual: detector y tracker

- **Detector (DETR-based):** procesa cada frame independientemente para encontrar nuevos objetos que coincidan con el concepto.
- **Tracker (herencia de SAM 2):** utiliza el mecanismo de memoria para propagar identidades temporalmente.
- **Sinergia:** el detector alimenta al tracker con nuevos objetos; el tracker mantiene la coherencia temporal. Si el tracker falla, el detector puede reiniciarlo.

Innovación clave: el Presence Token

- Soluciona el conflicto entre reconocer («¿qué es?») y localizar («¿dónde está?»).
- Introduce un token global que predice la probabilidad de que el concepto exista en la imagen, P(presence).
- Las queries de objetos solo se enfocan en la localización condicional, P(match | presence).
- Resultado: reduce drásticamente los falsos positivos en vocabulario abierto al distinguir entre la existencia y la posición del objeto.



Evolución: SAM → SAM 2 → SAM 3

La familia de modelos Segment Anything (SAM) ha evolucionado rápidamente, ampliando sus capacidades de segmentación universal desde imágenes estáticas hasta el análisis complejo de vídeo con prompts contextuales.

01

2023 · SAM 1

Segmentación universal en imagen con prompts de puntos y cajas. Dataset SA-1B: unos 11 millones de imágenes y 1.100 millones de máscaras.

02

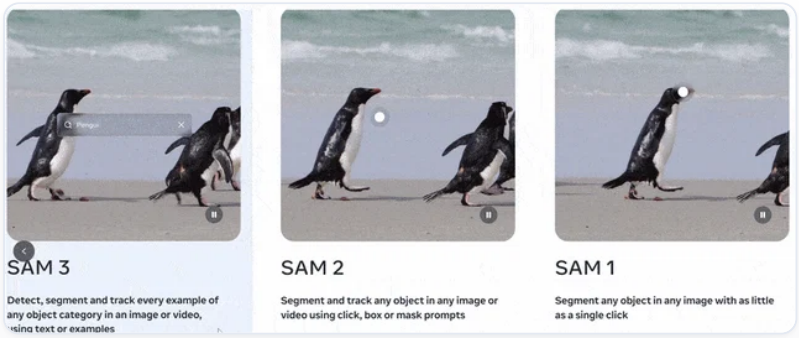
2024 · SAM 2

Extiende a vídeo con memoria en streaming; más preciso y hasta 6 veces más rápido que SAM 1 en imagen. Dataset SA-V: unos 51.000 vídeos y 643.000 masklets.

03

2025 · SAM 3

Modelo unificado que detecta, segmenta y sigue todas las instancias de un concepto, dado por texto o por ejemplos, en imagen y vídeo.



El motor de datos (data engine): cómo escalar a 1B de máscaras

La creación del dataset SA-1B, clave para el entrenamiento de SAM, se logró mediante un innovador proceso iterativo de «motor de datos», combinando la inteligencia humana con la capacidad de autoaprendizaje del modelo para escalar la anotación.

01

Anotación manual

Expertos humanos realizan anotaciones detalladas en un subconjunto inicial de imágenes. En esta fase, el modelo aún no tiene suficiente conocimiento.

02

Anotación semi-automática

El modelo, preentrenado con datos manuales, propone máscaras que los humanos revisan y corrigen. Esta interacción acelera el aprendizaje del modelo, que mejora exponencialmente.

03

Anotación totalmente automática

Una vez que el modelo es robusto, anota imágenes completas de forma autónoma. Solo se aceptan las máscaras con alta confianza, permitiendo un escalado masivo de datos.

Conclusión: SA-1B es 99,1 % generado automáticamente por el propio SAM, demostrando el poder del bucle de auto-mejora.

"Promptable segmentation": la idea central

La segmentación promptable es la capacidad de un modelo de visión artificial para segmentar objetos en imágenes o vídeos basándose en diferentes tipos de prompts (indicaciones). Esto permite una interacción flexible y eficiente sin necesidad de reentrenar el modelo.

Tipos de prompts y sus capacidades

Puntos

- **Múltiples puntos:** incluye o excluye áreas específicas para refinar la máscara deseada.
- **Precisión:** útil para objetos ambiguos o con bordes complejos.

Cajas (bounding box)

- **Delimitación amplia:** dibuja una caja alrededor del objeto; SAM la convierte en una máscara precisa.
- **Rapidez:** ideal para segmentación rápida de objetos bien definidos.

Máscara inicial

- **Refinamiento:** proporciona una máscara aproximada (rough) y el modelo perfecciona los bordes y rellena huecos.
- **Perfeccionamiento:** útil para mejorar segmentaciones previas o manuales.

Texto

- **SAM 1 y SAM 2:** mediante adaptadores (CLIP o Grounding DINO), se puede usar texto como «el coche rojo».
- **SAM 3 (nativo):** interpreta directamente prompts de texto («el árbol de la izquierda») para generar máscaras.

Ejemplar (crop)

- **SAM 3:** a partir de un recorte de ejemplo, segmenta todas las instancias del mismo concepto en la imagen o vídeo.
- **Generalización:** ideal para tareas de conteo o seguimiento de objetos idénticos.

Segment everything

- **Cobertura completa:** genera una máscara para cada objeto detectable en la imagen sin prompts específicos.
- **Análisis exhaustivo:** útil para la inspección completa de escenas complejas.

Meta AI publica un **Segment Anything Playground**, un espacio interactivo para probar estos prompts sobre medios propios.

La indicación decide qué sale

Un punto devuelve una instancia; un concepto en texto devuelve todas las que coinciden.

Máscaras precalculadas para esta demostración con un algoritmo clásico a partir de una caja. En el cuaderno las genera SAM.

Qué le indicas al modelo

- **Un punto en la sandía izquierda**
- Un punto en la sandía derecha
- Texto: «sandía»
- Una caja sobre las limas
- Un punto en la piña

Un punto señala una región concreta, así que el modelo devuelve esa sandía y no la otra.

Instancias segmentadas **1**

Esta página recoge el estado con un punto sobre la sandía izquierda. La diferencia que conviene retener no es visual sino de interfaz: un punto o una caja identifican *dónde*, y devuelven la instancia que hay ahí; un concepto en texto o un ejemplar identifican *qué*, y devuelven todas las coincidencias. SAM 1 y SAM 2 admiten de forma nativa lo primero, y para lo segundo necesitan un modelo externo que traduzca el texto; SAM 3 incorpora el texto y los ejemplares como prompts propios. Las máscaras de esta demostración se calcularon con GrabCut a partir de una caja, no con SAM: el cuaderno del tema las genera con el modelo real.

Qué devuelve exactamente una máscara

Desliza para quitar el fondo. Lo que queda es la salida de la segmentación: una lista de píxeles por instancia, cada una con su contorno, no un recorte rectangular.

Cuatro instancias con su contorno. Fíjate en las hojas de la piña y en el borde entre las dos filas de limas: ahí es donde se nota la calidad del recorte.

> Una máscara pesa poco y se guarda comprimida (RLE en COCO). Con ella puedes contar objetos, medir su área en píxeles o recortarlos con transparencia, cosas que una caja no permite.

La imagen de la derecha se compone multiplicando la foto por la unión de las cuatro máscaras, que es lo mismo que hace cualquier herramienta de recorte automático. Las máscaras están precalculadas con GrabCut a partir de una caja, no con SAM: el cuaderno del tema las genera con el modelo real. Un detalle que se ve al deslizar: el contorno de las limas rodea el montón entero porque la indicación fue una caja sobre el montón, no sobre una lima suelta.

Un clic elige el objeto

El punto positivo señala una región que debe pertenecer a la máscara.

Quieres segmentar una manzana entre otras frutas. ¿Qué prompt sirve para empezar?

- A Un punto positivo dentro de la manzana.
- B El nombre de la clase sin imagen.
- C Una métrica de precisión.

Respuesta A. Exacto: indica qué objeto quieres incluir.

La primera indicación puede admitir varias interpretaciones: objeto completo, parte o subparte. Por eso algunas versiones devuelven varias máscaras candidatas con puntuaciones.

Corregir sin volver a entrenar

Los nuevos clics añaden información sobre qué incluir y qué excluir.

La máscara incluye parte del fondo. ¿Qué harías?

- A Añadir un punto negativo sobre el fondo.
- B Subir la temperatura del modelo.
- C Cambiar el nombre del objeto.

Respuesta A. Exacto: el punto negativo señala qué región excluir.

En vídeo con SAM 2, una corrección puede actualizar la memoria y propagarse. La interacción debe evaluarse por calidad final y por número de clics o tiempo requerido.

Cuándo usar (y cuándo no) SAM

Úsalo si… (ideal para flexibilidad y zero-shot)

- **Segmentación zero-shot crítica:** necesitas segmentar objetos desconocidos, raros o que no existían en tu dataset de entrenamiento (por ejemplo microscopía o imagen submarina, underwater).
- **Vídeo y coherencia temporal:** requieres rastrear objetos que se mueven, ocultan y reaparecen sin entrenar un tracker específico, gracias a la memoria de SAM 2 y 3.
- **Interacción humana (human-in-the-loop):** estás construyendo herramientas de anotación de datos o edición donde el usuario refina el resultado con clics o cajas.
- **Búsqueda por conceptos (SAM 3):** necesitas encontrar todas las instancias de un concepto abstracto («tornillos oxidados») usando texto o ejemplos visuales, no solo puntos.
- **Escasez de datos:** no tienes miles de máscaras anotadas para entrenar un modelo supervisado tradicional.

Evítalo si… (mejor usar modelos supervisados clásicos)

- **Restricciones de hardware extremas (edge):** necesitas inferencia en milisegundos en dispositivos IoT o móviles antiguos. Aunque eficiente, SAM 3 requiere GPU potentes (H200) para latencias de unos 30 ms en cargas altas.
- **Entorno cerrado y repetitivo:** si tu problema es «detectar botellas en una cinta transportadora», con clases fijas y fondo controlado, un modelo ligero como RF-DETR-Seg será mucho más rápido y barato de desplegar.
- **Razonamiento complejo:** SAM 3 entiende frases nominales simples. Si necesitas lógica compleja, necesitas un VLM.

La máscara no nombra la especie

SAM delimita; otra señal debe reconocer, decidir o justificar.

Necesitas saber qué especie aparece y segmentarla.

- A Combinar una señal de categoría con SAM.
- B Usar solo una máscara sin revisar.
- C Eliminar los prompts.

Respuesta A. Exacto: SAM aporta la máscara, otra parte debe aportar la categoría.

SAM resulta valioso con poca anotación, interacción humana o vídeo. Un modelo supervisado pequeño puede ser más rápido y barato para clases estables y fondos controlados. Siempre hay que evaluar bordes, objetos pequeños, oclusiones y consumo de memoria.

Segmentar con un punto

Un clic, una máscara, sin entrenar nada para tu objeto.

```
import torch
from PIL import Image
from transformers import Sam2Model, Sam2Processor

model_id = "facebook/sam2.1-hiera-large"
model = Sam2Model.from_pretrained(model_id).eval()
processor = Sam2Processor.from_pretrained(model_id)

image = Image.open("mi_imagen.jpg").convert("RGB")
puntos = [[[450, 300]]]          # un punto, en coordenadas de la imagen
etiquetas = [[[1]]]              # 1 incluye, 0 excluye

inputs = processor(images=image, input_points=puntos,
                   input_labels=etiquetas, return_tensors="pt")
with torch.no_grad():
    salida = model(**inputs, multimask_output=True)

mascaras = processor.post_process_masks(salida.pred_masks,
                                        inputs["original_sizes"])[0]
print(mascaras.shape, salida.iou_scores)
```

- **multimask_output** devuelve tres candidatas para resolver la ambigüedad entre objeto, parte y subparte. Quédate con la de mayor `iou_scores` o enséñaselas al usuario.
- Para **corregir**, añade otro punto con etiqueta 0 donde la máscara invada el fondo y vuelve a llamar.
- El encoder de imagen es lo caro y se ejecuta una vez. Los clics posteriores son casi gratis.

¿Qué versión de SAM necesitas?

Las tres segmentan, pero no aceptan las mismas indicaciones ni el mismo medio.

¿Sobre qué trabajas?

- (●) Imágenes sueltas
- ( ) Vídeo, y hay que seguir el objeto

¿Cómo vas a señalar lo que buscas?

- (●) Con clics o cajas
- ( ) Escribiendo qué es, o con un recorte de ejemplo

SAM 1 basta

Imagen y prompts espaciales es justo su caso. Es el más sencillo de desplegar y el encoder se ejecuta una vez por imagen.

Si ya tienes SAM 2 montado, úsalo igualmente: en imagen también es más rápido.

SAM 2

Su banco de memoria mantiene la identidad del objeto entre fotogramas y aguanta oclusiones. Una corrección en un fotograma se propaga al resto.

`facebook/sam2.1-hiera-large`

SAM 3, o SAM 2 con un detector delante

SAM 3 acepta texto y ejemplares de forma nativa y devuelve todas las instancias del concepto. La alternativa clásica es Grounding DINO produciendo cajas que alimentan a SAM 2.

Esa combinación es la que monta el cuaderno del tema.

SAM 3

Es el único que une las dos cosas: detecta por concepto y sigue las instancias en el tiempo, con el detector reiniciando al tracker cuando lo pierde.

Cuenta con que es el más exigente en hardware de los tres.

Antes de montar SAM en un flujo real

Seis comprobaciones sobre bordes, coste e interacción.

- [ ] **Bordes difíciles probados**
  Pelo, transparencias, objetos muy pequeños. Ahí es donde se cae la máscara.
- [ ] **Ambigüedad resuelta**
  Decide si te quedas con la máscara de mayor puntuación o le enseñas las tres al usuario.
- [ ] **Coste de interacción medido**
  Cuántos clics de media hacen falta para una máscara aceptable.
- [ ] **Encoder reutilizado**
  Una vez por imagen, no una por clic. Es la diferencia entre fluido y lento.
- [ ] **Reconocimiento resuelto aparte**
  SAM delimita pero no nombra. Decide quién pone la etiqueta.
- [ ] **Alternativa ligera comparada**
  Con clases fijas y fondo controlado, un modelo supervisado pequeño puede salir mejor y más barato.

Segmenta y corrige

Prueba cómo cambian las máscaras con puntos positivos y negativos.

1. Guarda una copia en Drive.
2. Segmenta con un punto.
3. Refina y compara máscaras.

Abrir en Colab ↗

Empieza con un punto positivo y añade un punto negativo donde la máscara invada el fondo. Compara número de interacciones, contorno y objetos pequeños. Comprueba qué versión carga el cuaderno: texto, memoria temporal y ejemplares no están disponibles en todos los modelos SAM.