

# Instrucciones **Tracker** con SpeciesNet + Docker (CPU/GPU)

9 de noviembre de 2025

## Resumen

Cómo ejecutar el *Caracal Tracker* basado en **SpeciesNet** dentro de Docker para:

- procesar un video completo y generar MP4 anotado y CSV (-t);
- inspeccionar un frame al % del video (-f <0-100>);
- ejecutar en **CPU** o **GPU** con docker compose.

## Requisitos

- Docker Desktop actualizado.
- Proyecto clonado localmente.
- (GPU) Drivers NVIDIA y soporte de CUDA en el host. Verificación rápida:

```
docker run --rm --gpus all --entrypoint nvidia-smi nvidia/cuda:12.5.1-base-ubuntu22.04
```

## Estructura del proyecto

```
SpeciesNet_version/
videos/
    caracal_video_02.mp4
outputs/
run_speciesnet.py
speciesnet_tracker.py
Dockerfile.cpu
Dockerfile.gpu
docker-compose.cpu.yml
docker-compose.gpu.yml
```

## Build de imágenes

Ubicarse en la raíz del proyecto:

```
cd SpeciesNet_version
```

### CPU

```
docker compose -f docker-compose.cpu.yml build --no-cache
```

### GPU

El archivo `docker-compose.gpu.yml` usa `gpus: - all` (lista). Construir:

```
docker compose -f docker-compose.gpu.yml build --no-cache
```

## Ejecución (tracking completo)

### CPU

```
docker compose -f docker-compose.cpu.yml run --rm caracal-tracker-cpu \
-t --video videos/caracal_video_02.mp4 \
--min-conf 0.15 --batch-size 24 --stride 1 \
--primary-only --lag 6 --min-persist 8 --max-gap 12 --startup-suppress 15
```

### GPU

```
docker compose -f docker-compose.gpu.yml run --rm caracal-tracker-gpu \
-t --video videos/caracal_video_02.mp4 \
--min-conf 0.15 --batch-size 48 --stride 1 \
--primary-only --lag 6 --min-persist 8 --max-gap 12 --startup-suppress 15
```

**Notas.** `-stride` controla cada cuántos frames se detecta (1 = todos). `-batch-size` alto favorece GPU. `-primary-only` dibuja sólo el track principal (caja azul estable). `-lag`, `-min-persist`, `-max-gap` y `-startup-suppress` suavizan y estabilizan la caja usando una ventana temporal.

### Inspección de un frame al %

```
docker compose -f docker-compose.gpu.yml run --rm caracal-tracker-gpu \
-f 5 --video videos/caracal_video_02.mp4 --min-conf 0.15
```

### Salidas

- MP4 anotado: `outputs/out_speciesnet.mp4`
- Log CSV: `outputs/speciesnet_log.csv`
- Inspección de frame: `outputs/frame_inspect/frame_X.jpg`
- Tiempo total de detección (seg.): impreso al final como `DET_TIME_SEC=...`

### Parámetros útiles

- `-t`: ejecuta el tracking completo del video.
- `-f <0-100>`: inspecciona el frame correspondiente al porcentaje indicado del video.
- `-video <ruta>`: ruta del archivo de video de entrada (usar barras / o comillas en Windows).
- `-min-conf <float>`: umbral mínimo de confianza del detector (ej. 0.15).
- `-batch-size <int>`: tamaño de lote para el detector (valores mayores favorecen GPU; p. ej. 24 en CPU, 48 en GPU).
- `-stride <int>`: detecta cada N frames (1 = todos los frames).
- `-primary-only`: dibuja sólo la caja del track principal (suavizada y estable).
- `-lag <int>`: retraso en frames para estabilizar la caja (look-ahead); mayor = más estable pero más delay.
- `-min-persist <int>`: frames mínimos que un track debe persistir antes de considerarse principal.
- `-max-gap <int>`: frames máximos que se mantiene el track principal sin detecciones (con extrapolación suave).
- `-startup-suppress <int>`: durante los primeros N frames filtra detecciones débiles para evitar falsos positivos iniciales.
- `-max-frames <int>`: limita la cantidad total de frames a procesar.
- `-start-frame <int>`: índice del frame inicial desde el que comenzar a procesar.

- `-save-max-side <int>`: redimensiona el frame para el detector según lado máximo (mejora rendimiento; p. ej. 960).
- `-jpeg-quality <40-100>`: calidad JPEG temporal al guardar frames para el detector.

## Ciclo de desarrollo

Si sólo cambian archivos .py, no es necesario recomilar: basta con volver a ejecutar los comandos de `run`. Reconstruir sólo si cambian `Dockerfile*` o dependencias.

## Solución de problemas

- “**gpus must be a list**”: en Compose usar

```
gpus:
  - all
```

- **No abre el video / ruta en Windows**: usar barras / o comillas:

```
--video "videos/caracal_video_02.mp4"
```

- **Sin detecciones al inicio**: aumentar `-startup-suppress` o bajar `-min-conf`.

- **Múltiples cajas inestables**: usar `-primary-only` y ajustar `-lag`, `-min-persist`, `-max-gap`.

- **Limpiar contenedores huérfanos**:

```
docker compose --project-name speciesnet_version down --remove-orphans
```

- **Probar GPU dentro del contenedor**:

```
docker compose -f docker-compose.gpu.yml run --rm \
  --entrypoint nvidia-smi caracal-tracker-gpu
```