

AnimalDet: Sistema de Detección Multiespecie mediante RF-DETR

para Imágenes Aéreas de la Sabana Africana

Amir Sadour

MaIA

Universidad de los Andes

a.sadour@uniandes.edu.co

Camilo Rodriguez

MaIA

Universidad de los Andes

ca.rodriguezs123@uniandes.edu.co

Claudia Agudelo

MaIA

Universidad de los Andes

cj.agudelo@uniandes.edu.co

Luis Manrique

MaIA

Universidad de los Andes

l.manriquec@uniandes.edu.co

1. Demo en vivo

Puede ver una demostración en este enlace

2. Repositorio

El código puede encontrarse en github

3. Guía general de instalación de Animaldet

Animaldet fue construido como una aplicación contenerizada que puede ejecutarse localmente o en cualquier nube que soporte contenedores

4. Dependencias y Requisitos del Sistema

4.1. Requisitos de Hardware

- **Memoria RAM:** Mínimo 8 GB (recomendado 16 GB)
- **Espacio en disco:** 10 GB disponibles
- **GPU (opcional):** NVIDIA con soporte CUDA 11.8+ para inferencia acelerada
- **CPU:** Procesador multi-core (mínimo 4 cores)

4.2. Software Requerido

Para ejecución con Docker (Recomendado):

- **Docker Desktop** (versión 20.10 o superior) o Docker Engine en Linux
- **Git** (versión 2.0 o superior)
- **Git LFS** para descarga de modelos

Para desarrollo local:

- **Python:** 3.12 o superior
- **Node.js:** 20.x o superior (para compilar frontend)
- **uv:** Gestor de paquetes de Python

4.3. Dependencias de Python

El proyecto utiliza las siguientes librerías principales:

- **FastAPI:** Framework web para la API
- **ONNX Runtime:** Motor de inferencia optimizado
- **Pillow:** Procesamiento de imágenes
- **NumPy:** Computación numérica
- **Uvicorn:** Servidor ASGI

Ver archivo `pyproject.toml` para lista completa de dependencias.

4.4. Dependencias del Frontend

- **React:** 18.x - Framework UI
- **Vite:** Build tool y dev server
- **TypeScript:** Tipado estático
- **TailwindCSS:** Framework de estilos

Ver archivo `ui/package.json` para lista completa.

5. Despliegue local

Primero, con el repositorio clonado debe ejecutar

```
git lfs pull
```

Para obtener los modelos y otra información útil para desplegar su modelo.

Debe tener **docker** instalado.

Después de eso puede ejecutar

```
make build  
docker run -p 8000:8000 animaldet:latest
```

Luego al abrir en su navegador `localhost:8000` debe ver la aplicación funcionando.

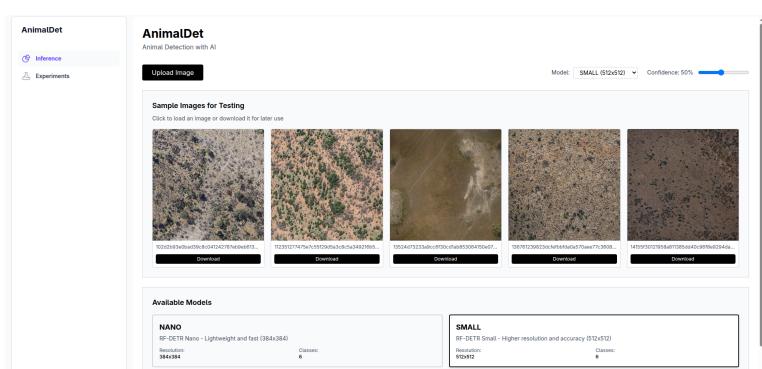


Figura 1: Vista principal

Para la guía de usuario sobre cómo usar Animaldet puede consultar este enlace

6. Despliegue en la nube

La plataforma de despliegue incluida en Animaldet es **AWS ECS**, con la siguiente arquitectura:

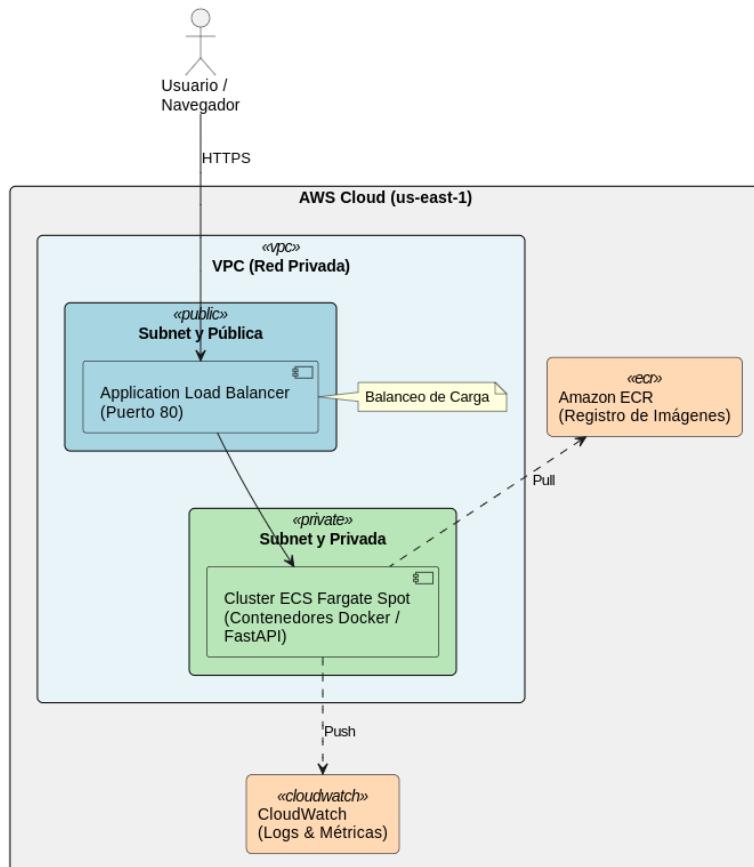


Figura 2: Arquitectura

Incluimos toda la infraestructura requerida como código en `infra/aws`, donde usamos `terraform`.

Para desplegar la aplicación debe configurar sus credenciales de AWS, más información en documentación de configuración de AWS CLI

Comandos para desplegar la aplicación

```
make deploy
```

Esto ejecutará `docker build`, luego `terraform apply` que va a:

1. Construir la imagen Docker localmente
2. Iniciar sesión en el registro ECS
3. Subir la imagen a ECS
4. Crear clusters, definiciones de tareas, políticas y balanceador de carga para que la aplicación funcione

7. Parametrización

El sistema permite ajustar parámetros de funcionamiento mediante variables de entorno:

Variable	Valor por defecto	Descripción
MODEL_PATH	model.onnx	Ruta al modelo ONNX
CONFIDENCE_THRESHOLD	0.5	Umbral de confianza (0.0-1.0)
NMS_THRESHOLD	0.45	Umbral de NMS
RESOLUTION	512	Resolución de entrada
USE_STITCHER	true	Habilitar stitching
PORT	8000	Puerto HTTP

Cuadro 1: Variables de entorno configurables

7.1. Ejemplo de configuración

Opción 1: Usando variables de entorno

```
docker run -p 8000:8000 \
-e CONFIDENCE_THRESHOLD=0.6 \
-e RESOLUTION=640 \
animaldet:latest
```

Opción 2: Usando archivo .env

Crear archivo .env en la raíz del proyecto:

```
MODEL_PATH=model.onnx
CONFIDENCE_THRESHOLD=0.5
RESOLUTION=512
USE_STITCHER=true
PORT=8000
```

Ejecutar con:

```
docker run -p 8000:8000 --env-file .env animaldet:latest
```

8. Animaldet: cómo usar nuestra herramienta de detección de animales

Primero, la pantalla principal:

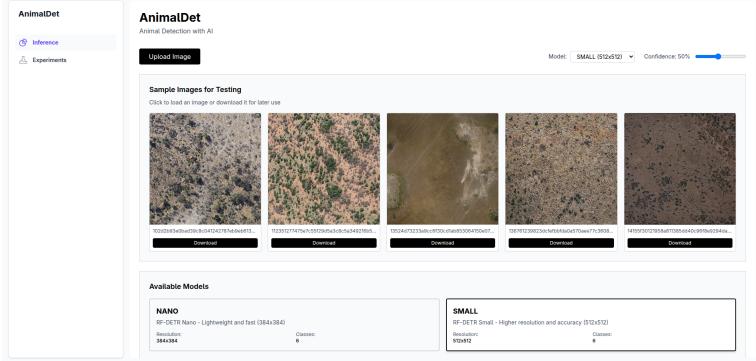


Figura 3: Pantalla principal

Puede elegir sus propias imágenes o seleccionar una de las muestras.

Si decide seleccionar una de las muestras, verá las anotaciones de la imagen y un botón para ejecutar predicciones:

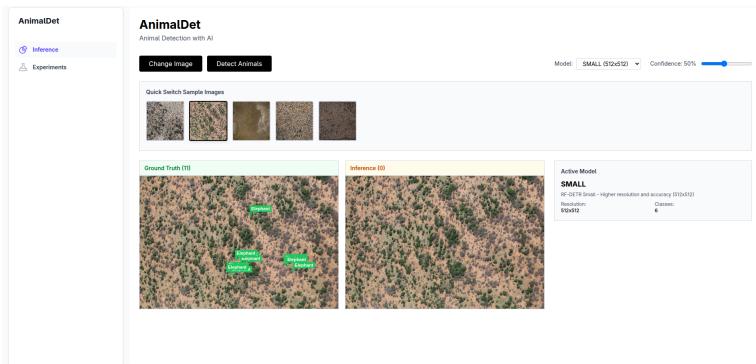


Figura 4: Muestra seleccionada

También puede elegir entre el modelo **small** y **nano**, y cambiar la confianza del modelo.

8.1. Tamaño del modelo

El modelo Nano usa 30 % menos RAM que el modelo Small, pero como usa una resolución menor debe realizar más inferencias sobre toda la imagen, los tiempos de inferencia no difieren mucho entre estos modelos.

8.2. Confianza

Cada modelo tiene su propia confianza sugerida, la confianza es una salida del modelo que nos indica qué tan seguro está sobre una caja delimitadora encontrada con su respectiva clase, su confianza recomendada es la que maximiza el rendimiento pero puede experimentar con diferentes valores

8.3. Realizando inferencia

Una vez que la imagen y los parámetros están elegidos, puede hacer clic en el botón Detect Animals y comenzará una pantalla de carga

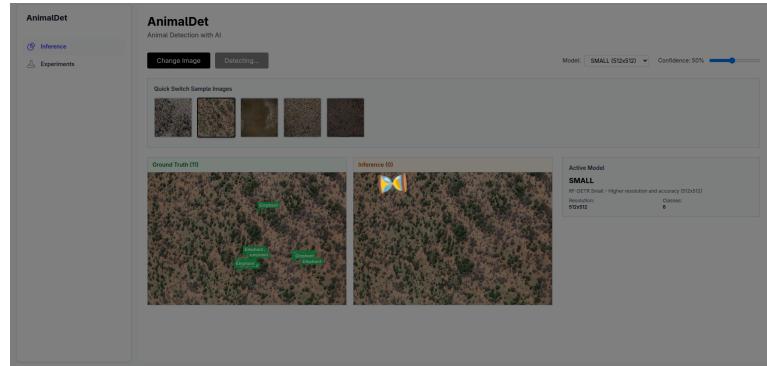


Figura 5: Pantalla de carga

Una vez finalizado mostrará las detecciones de la imagen:

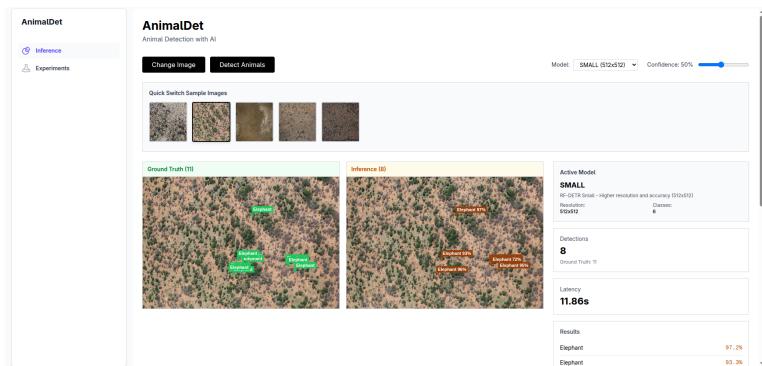


Figura 6: Detecciones

9. Confianza baja: sobre-detección

Cuando elige una confianza más baja puede esperar sobre-predicción como se muestra en la siguiente captura:

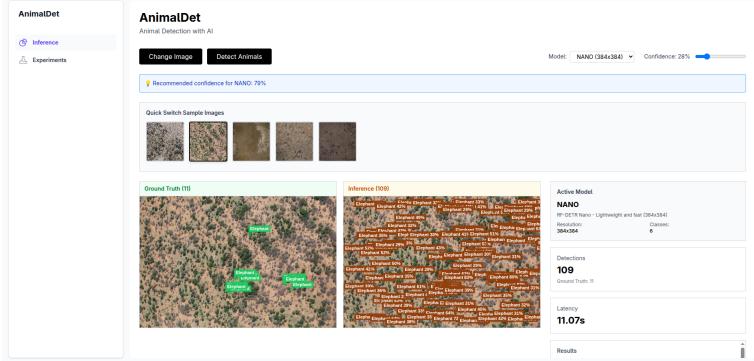


Figura 7: Sobre-detección

Puede funcionar para propósitos de experimentación, pero por favor tenga cuidado.

10. Comparación de experimentos: veamos cómo se desempeñó el modelo

Hemos agregado una vista de experimentos donde puede ver detecciones sobre todo el conjunto de datos de prueba comparado con el ground truth:

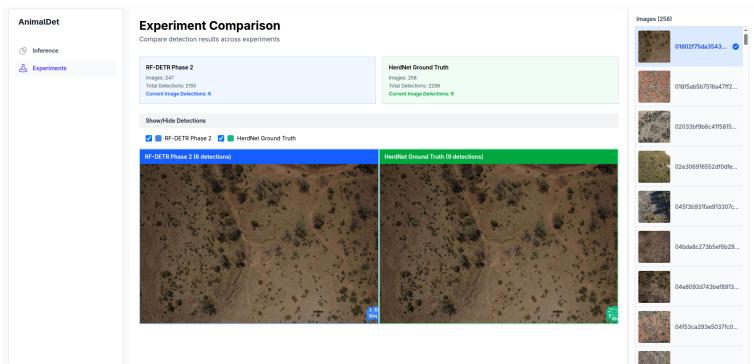


Figura 8: Vista de experimentos

Allí puede elegir cualquier imagen del conjunto de prueba y verificar la diferencia entre el ground truth y sus predicciones.