



**Instituto Tecnológico de Estudios Superiores de Monterrey,
Campus Querétaro**

Manual

Manual de Instalación y Despliegue de CowVision

Presenta

Carlos Rodrigo Salguero Alcántara

A00833341

Diego Perdomo Salcedo

A01709150

CowVision: Guía de Instalación y Despliegue	2
Introducción	2
Características Principales	2
Requisitos del sistema	2
Instalación	2
Ejecutando Inferencia (Predicciones)	3
Procesamiento de Imágenes Individuales	3
Ejemplo de salida con información visual	4
Ejemplo de salida sin información visual	4
Procesamiento de Múltiples Imágenes	4
Configuración avanzada	5
Entrenamiento del Modelo	5
Entrenamiento en Computadora sin GPU	6
Solución de Problemas	7
Entrenamiento en Computadoras más Potentes	7
Consejos Generales	8

CowVision: Guía de Instalación y Despliegue

Introducción

CowVision es un sistema avanzado de visión por computadora diseñado para detectar vacas utilizando modelos YOLO (You Only Look Once).

Características Principales

- ❖ 🔍 **Detección Precisa:** utiliza YOLO9 para detectar vacas con alta precisión.
- ❖ 📊 **Procesamiento Eficiente:** capaz de procesar imágenes individuales o directorios completos.
- ❖ ⚙️ **Personalizable:** ajusta los parámetros de detección según tus necesidades.
- ❖ 🍓 **Compatible con Raspberry Pi:** optimizado para funcionar en hardware de bajo consumo

Requisitos del sistema

- ❖ Raspberry Pi 4 (recomendado) o superior
- ❖ Al menos 4GB de RAM
- ❖ Python 3.10
- ❖ Espacio libre de almacenamiento de al menos 2GB
- ❖ Sistema operativo Raspberry Pi OS o Ubuntu 22.04

Instalación

Sigue estos pasos para instalar CowVision en tu Raspberry Pi:

1. Actualiza tu sistema

```
sudo apt update && sudo apt upgrade -y
```

2. Instala las dependencias necesarias (en caso de no contar con ellas)

```
sudo apt install -y python3-pip python3-venv libopencv-dev
```

3. Descarga la última versión de CowVision

```
# Clona el repositorio o descarga el archivo zip de la versión
git clone https://github.com/salgue441/caetec-cow-classification
cd caetec-cow-classification
```

4. Crea un entorno virtual de Python

```
python3 -m venv {nombre}
source {nombre}/bin/activate
```

5. Instala el paquete en modo desarrollo

```
pip install -e .
```

6. Descarga y coloca los archivos del modelo

Descarga los archivos del modelo desde [Drive](#) o la sección de lanzamiento (releases) en GitHub.

- ❖ Crea una carpeta llamada `model` a nivel raíz del proyecto
- ❖ Coloca `yolov9c.pt` en el directorio `model`.
- ❖ Coloca `best.pt` en el directorio `model`.

```
mkdir -p model
```

Ejecutando Inferencia (Predicciones)

Una vez instalado, puedes utilizar CowVision para detectar varias vacas en imágenes. La herramienta principal es `run-cow-inference`.

Procesamiento de Imágenes Individuales

Para procesar una sola imagen y guardar el resultado con cajas de detección:

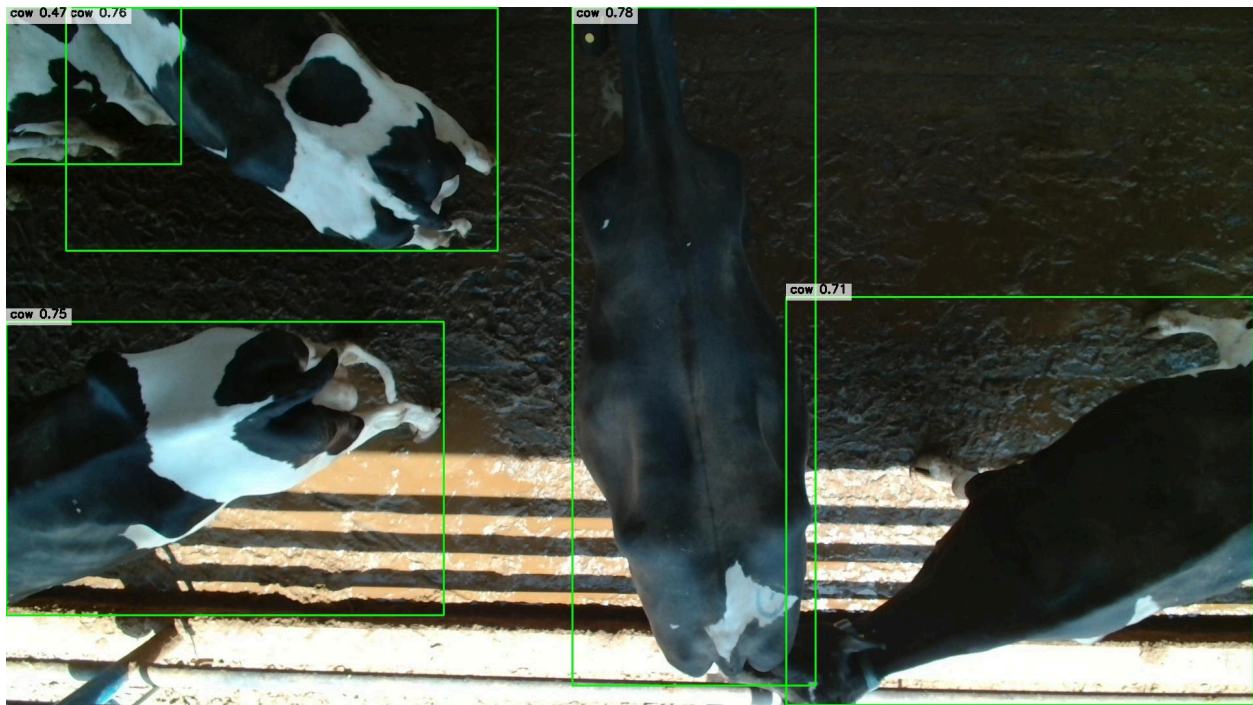
```
# Activa el entorno virtual primero
```

```
source {nombre}/bin/activate
```

```
# Ejecuta la inferencia en una imagen con salida visual
```

```
Run-cow-inference --image ruta/a/imagen.jpg --output  
resultados/detectada.jpg
```

Ejemplo de salida con información visual



Para obtener solo la información de detección sin salida visual

```
run-cow-inference --image ruta/a/imagen.jpg --no-render
```

Ejemplo de salida sin información visual

Image: 2024-04-12-14-30-03_jpg.rf.230619284902124f1e5aa940f3b79ef9.jpg

Detections: 5

Detection 1: probability 0.78

Detection 2: probability 0.76

Detection 3: probability 0.75

Detection 4: probability 0.71

Detection 5: probability 0.47

Procesamiento de Múltiples Imágenes

Para procesar todas las imágenes en un directorio

```
run-cow-inference --dir ruta/a/imagenes/ --output resultados/
```

El directorio de salida se creará si no existe. Cada imagen procesada se guardará con “_detected” añadido al nombre del archivo.

Configuración avanzada

Puedes ajustar los parámetros de detección para mejorar los resultados

```
# Ajusta los umbrales de confianza e IoU
run-cow-inference --image vacas.jpg \
                  --output detectadas.jpg \
                  --conf 0.7 \
                  --iou 0.5 \
                  --min-area 400
```

Explicación de parámetros:

- ❖ **--conf** es el Umbral de confianza (0.0 - 1.0). Valores más altos requieren mayor certeza.
- ❖ **--iou** es el umbral de Intersección sobre Unión (0.0 - 1.0). Controla la eliminación de duplicados.
- ❖ **--min-area** es el área mínima (en píxeles) para considerar una detección válida.

Entrenamiento del Modelo

Si necesitas reentrenar el modelo con tus propios datos, sigue estos pasos:

1. **Prepara tu conjunto de datos:** organiza tus imágenes (etiquetadas con bounding-boxes) y anotaciones en el siguiente formato:

```
data/
├── images/
│   ├── train/
│   ├── val/
│   └── test/
├── labels/
│   ├── train/
│   ├── val/
│   └── test/
└── data.yaml
```

El archivo `data.yaml` debe contener:

```
train: data/images/train/
val: data/images/val/
test: data/images/test/
nc: 1 # número de clases
names: ['cow'] # nombre de las clases
```

2. Inicia el entrenamiento

```
# Activa el entorno virtual
source {nombre}/bin/activate

# Ejecuta el entrenamiento
train-cow-model --data_yaml data/data.yaml --epochs 100 \
                --batch_size 16
```

Parámetros importantes:

- ❖ `--epochs` es el número de épocas de entrenamiento (más épocas = más tiempo de entrenamiento)

❖ `--batch_size` es el tamaño del lote (reducir si hay problemas de memoria)

3. **Monitoreo del entrenamiento:** el progreso del entrenamiento se guardará en la carpeta `runs/`. Puedes revisar las métricas y visualizaciones generadas.
 - a. Curva de pérdida y precisión
 - b. Imágenes de validación con predicciones
 - c. Matrices de confusión
 - d. Métricas de rendimiento (mAP, precisión, recall)
4. **Usando el modelo entrenado:** una vez completado el entrenamiento, el modelo guardará los mejores y últimos pesos en `runs/detect/trainX/weights/best.pt` y `runs/detect/trainX/weights/last.pt`. Puedes usar estos pesos para inferencia.

Entrenamiento en Computadora sin GPU

Si no tienes acceso a una GPU, aún puedes entrenar en CPU, pero será mucho más lento. Considera reducir el tamaño de las imágenes y el número de épocas para hacer el entrenamiento más manejable en CPU.

Solución de Problemas

Problema	Solución
<code>FileNotFoundError</code>	Verifica que los archivos del modelo estén ubicados correctamente en la carpeta de <code>modelo</code> .
Errores de importación	Asegúrate de haber activado el entorno virtual.
<code>ImportError: No module named ...</code>	Verifica que todas las dependencias estén instaladas <code>pip install -e .</code>
Errores de memoria	Reduce el tamaño de las imágenes o ajusta el batch size durante el entrenamiento.
Rendimiento lento en inferencia (predicciones)	Usa imágenes más pequeñas y considera ajustar el parámetro <code>--conf</code> entre 0.3 y 0.4.
Error “CUDA not available”	Si no tienes GPU, el modelo automáticamente usará CPU. Ignora este mensaje.
Error “CUDA out of memory”	Reduce el batch size durante el entrenamiento o usa imágenes más pequeñas.
Problemas con OpenCV	Asegúrate de tener instaladas las dependencias correctas: <code>sudo apt install libopencv-dev.</code>
No se detectan objetos	Intenta reducir el umbral de confianza <code>--conf 0.2</code> para aumentar la sensibilidad.
Detecciones falsas	Aumentar el umbral de confianza <code>--conf 0.6</code> para mejorar la precisión.

Entrenamiento en Computadoras más Potentes

Para entrenar el modelo en una computadora de escritorio o servidor con más recursos (recomendado), sigue estos pasos:

1. Configura el entorno de tu computadora

```
python3 -m venv {nombre}
```

```
source {nombre}/bin/activate
```

```
pip install -e .
```

2. Aprovecha la GPU si está disponible

- ❖ Si tienes una GPU de NVIDIA, instala CUDA y cuDNN según las instrucciones oficiales
- ❖ Instala la versión de PyTorch compatible con CUDA

```
pip install torch torchvision --index-url  
https://download.pytorch.org/whl/cu118
```

3. Ajusta los parámetros de entrenamiento para hardware potente

```
train-cow-model -data_yaml data/data.yaml -epochs 300 -batch_size 16
```

4. Transfiere el modelo entrenado a la Raspberry Pi:

- ❖ Copia el archivo de pesos `runs/detect/train/weights/best.pt` a tu Raspberry Pi.
- ❖ Colócalo en la ruta correcta: `models/best.pt`.

Consejos Generales

1. Rendimiento óptimo

- Usa el formato de imagen JPG en lugar de PNG para un procesamiento más rápido
- Para entrenamiento, usa una GPU si es posible; para inferencia, la CPU es suficiente en muchos casos

2. Gestión de datos

- Organiza tus imágenes de entrenamiento de tus modelos entrenados
- Haz copias de seguridad de tus modelos entrenados
- Considera usar un servicio de almacenamiento en la nube para datasets grandes

3. Optimización de flujo de trabajo

- a. Entrena en hardware potente, despliega para inferencia en dispositivos más pequeños
- b. Monitorea el rendimiento de tu modelo con diferentes configuraciones
- c. Documenta los parámetros y resultados de cada entrenamiento