Workflow: YOLOv8 Hornet/Bee Detection on Raspberry Pi

Lucas Aurouet

December 15, 2025

1. Save the Trained Model

```
from ultralytics import YOLO

# Load base model
model = YOLO("yolov8n.pt")

# Fine-tuning
model.train(data="data.yaml", epochs=15, imgsz=640)

# Save weights
model.save("finetuned_weights.pt")
```

---

2. Export Model to ONNX

```
# Export to ONNX
model.export(format="onnx", opset=12, dynamic=True)
# Generates 'model.onnx'
```

---

3. Prepare Raspberry Pi

```
sudo apt update
sudo apt install python3-pip libatlas-base-dev
pip3 install numpy opencv-python onnxruntime
```

Enable Pi Camera in `raspi-config` if needed.

—

4. Copy Project to Pi

```
scp -r yolov8-raspberry-pi pi@<IP_DU_PI>:/home/pi/
```

Include scripts and model files.

—

5. Inference Script (Camera Input)

```
import cv2
import numpy as np
import onnxruntime as ort

# Load ONNX model
session = ort.InferenceSession("model.onnx")
input_name = session.get_inputs()[0].name

# Open camera
cap = cv2.VideoCapture(0)
cap.set(cv2.CAP_PROP_FRAME_WIDTH, 640)
cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)

while True:
    ret, frame = cap.read()
    if not ret:
        break

    # Preprocess
    img_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    img_resized = cv2.resize(img_rgb, (640, 640))
    img_float = img_resized.astype(np.float32) / 255.0
    input_tensor = np.expand_dims(np.transpose(img_float
       , (2,0,1)), axis=0)

    # Inference
    outputs = session.run(None, {input_name:
       input_tensor})
```

```
        # Post-processing
        for box in outputs[0]:
            x1, y1, x2, y2, conf, cls = box
            cv2.rectangle(frame, (int(x1), int(y1)), (int(x2
                ), int(y2)), (0,255,0), 2)
            cv2.putText(frame, f"{int(cls)}:{conf:.2f}",
                        (int(x1), int(y1-5)),
                        cv2.FONT_HERSHEY_SIMPLEX, 0.5,
                           (0,255,0), 1)

        cv2.imshow("Detection", frame)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

cap.release()
cv2.destroyAllWindows()
```

—

6. Run Script on Pi

```
cd ~/yolov8-raspberry-pi
python3 detect_frames.py
```

- Press `q` or `Ctrl+C` to stop. - Script captures camera frames, runs inference, and shows boxes in real time.

—

7. Optimization Tips

- Reduce image size (e.g., 320x320) for higher FPS.
- Use FP16 or INT8 quantization for ONNX.
- Use threading to separate capture and inference.
- On Pi 5 + Hailo, use Hailo runtime for acceleration.