



Escuela de Ingeniería en Electrónica
Taller de Sistemas Embebidos

Bitácora de Trabajo

Jean Carlos Chacón Pérez
jk@estudiantec.cr
2018093684

Cartago, Costa Rica
Noviembre 2025

1. Bitácora de Trabajo

Registro de Actividades

- **2025-11-01:** Se revisó el documento de orientación del Proyecto 2 de Taller de Sistemas Embebidos para comprender la rúbrica de evaluación, los requerimientos funcionales y los entregables. Se definió la estructura del proyecto, la división de roles y las responsabilidades del equipo.
- **2025-11-02:** Se elaboró la propuesta de diseño con todos los apartados solicitados en la rúbrica: hardware, software base, modelo de inteligencia artificial, pipeline técnico y despliegue en Yocto. Se documentaron los parámetros de desempeño esperados, precisión, FPS mínimo, latencia y estabilidad.
- **2025-11-05:** Se configuró el entorno de desarrollo en Windows, creando un entorno virtual con `venv` y se instalaron las librerías necesarias: `opencv-python`, `numpy`, `pyyaml`, `requests` y `tensorflow`. Se corrigieron errores de importación iniciales (No module named `tensorflow`, `tflite_runtime`, etc.).
- **2025-11-06:** Se construyó la estructura base del proyecto `yolo_tflite_starter/`, incluyendo los directorios `models/`, `data/` y `output/`. Se redactó el archivo de configuración `config.yaml` y el script `detect_yolo_tflite.py`, el cual soporta entrada por cámara o video, grabación automática de resultados y envío de datos a ThingSpeak.
- **2025-11-07:** Se instaló y configuró el repositorio oficial de YOLOv5. Se exportó el modelo `yolov5n.pt` al formato TensorFlow Lite mediante el comando `python export.py --weights yolov5n.pt --include tflite`. El modelo resultante `yolov5n-fp16.tflite` fue integrado al sistema.
- **2025-11-08:** Se realizaron las primeras pruebas de inferencia con el modelo en formato TFLite, utilizando videos MP4 locales y la cámara. Se resolvieron errores relacionados con rutas en YAML, tipos de salida del modelo y errores de empaquetado (`ValueError: too many values to unpack`).
- **2025-11-08:** Se corrigió el método de post-procesamiento del modelo YOLOv5 para interpretar correctamente la salida cruda (`[x, y, w, h, obj, cls0, cls1, ...]`), calculando la clase con mayor probabilidad y el puntaje combinado. Con esto se obtuvo inferencia estable y detecciones correctas.
- **2025-11-09:** Se verificó el funcionamiento del sistema con un video de prueba, confirmando la generación automática de un video procesado con detecciones. Se implementó y probó la función de **Non-Max Suppression (NMS)** en NumPy para eliminar cajas superpuestas, mejorando significativamente la visualización.

- **2025-11-09:** Se activó la opción de captura en vivo (`--source 0`) para el uso de cámara. Se comprobó el correcto funcionamiento del flujo de video en tiempo real, el cálculo de FPS y latencia por cuadro.
- **2025-11-10:** Se configuró y probó la integración con la API de **ThingSpeak**, usando la ruta `https://api.thingspeak.com/update`. Se validó la publicación de mensajes con los conteos de objetos detectados (peatones, vehículos, bicicletas, etc.) mediante una clave de escritura (**Write API Key**) en el campo `field1`.
- **2025-11-10:** Se realizaron pruebas de envío periódico (cada 15 segundos) a ThingSpeak y se verificó la correcta visualización de los datos en la interfaz web del canal. Se documentó el formato de los mensajes enviados y la estructura final del sistema.
- **2025-11-11:** Se finalizó la versión estable del script, con los parámetros ajustables de confianza (`--score`) e IoU (`--iou`). Se completó la bitácora de desarrollo y se validó la ejecución integral: detección, grabación, visualización y envío de datos en tiempo real.