Turorial guiado: TensorFlow Lite **Milton Murillo Vega**
Taller de Sistemas Embebidos Código: EL5841
Prof. Ing. Johan Carvajal Godínez Fecha: 27/10/2025
Escuela de Ingeniería en Electrónica Instituto Tecnológico de Costa Rica

Introducción

Esta guía corresponde a la ejecución de un modelo de TensorFlow Lite que clasifica razas de perros. EL procedimiento consiste en correr el modelo mediante un script de python con una imagen en formato .jpg en la que aparezca un perro para lograr determinar su raza. Seguido de ello, construir una imagen mínima utilizando Buildroot para la implementación del modelo y probar su eficiencia.

Implementación

A continuación se presentan los pasos para la implementación y ejecución del modelo utilizando Python.

1. Paso. Ambiente de trabajo.
   Clone el repositorio en donde trabajará el minitaller mediante el comando siguiente.

   ```
   git clone https://github.com/Milton-Murillo/Minitaller_TensorFlowLite.git
   ```

2. Paso. Descargue una imagen de un perro en formato **.jpg** y reemplace por la incluida. Documente la imagen en resultados, en la descripción mencione la raza del perro.

3. Paso. Ejecutar el script de Python. Dentro del directorio que almacena los archivos, abrir una terminal y ejecutar el script con el siguinte comando.

   ```
   python3 classify.py
   ```

   Tome una captura del resultado de detección y documente en resultados; en la descripción coloca **Resultados de la ejecución en mi computador**.

4. Paso. Crear una carpeta de overlay.
   Para un ambiente de compatible con el script, es necesario hacer una capa que se superpone a la estructura base de Buildroot, y una distribución de paquetes precompilados para Python. Ejecuta el comando en la ruta principal de trabajo.

   ```
   mkdir -p overlay/opt/wheels
   mkdir -p overlay/opt/app
   pip download --only-binary=:all: \
     --platform manylinux2014_x86_64 \
     --python-version 3.11 \
     --abi cp311 \
     numpy==1.26.4 pillow==10.4.0 \
     -d overlay/opt/wheels

   pip download --only-binary=:all: \
     --platform manylinux2014_x86_64 \
   ```

```
  --python-version 3.11 \
  --abi cp311 \
  tflite-runtime==2.14.0 \
  -d overlay/opt/wheels
```

5. Paso. Almacenar los archivos necesarios en la ruta **overlay/opt/app**.

   Mueve los archivos **classify.py**, **model.tflite**, **labels.txt** e **imagen.jpg** al directorio.

6. Paso. Clonar el ambiente de Buildroot.

```
git clone https://gitlab.com/buildroot.org/buildroot.git
cd buildroot
git checkout 2023.11.2
```

7. Paso. Configurar Buildroot.
   Desde la carpeta **Buildroot** ejecutar el comando a continuación.

```
make qemu_x86_64_defconfig
make menuconfig
```

   Dirigirse a Toolchain y en C library asegurarse de que esté la opción glibc, luego buscar y marcar la opción [*] Enable C++ support.

   Volver al menú principal y dirigirse a Target packages buscar la opción Interpreter languages and scripting y seleccionar python3 = [*], dentro de python3 entrar a External Python modules; buscar y marcar pip = [*], python-numpy = [*] y python-pillow = [*].

   Volver al menú principal y dirigirse a System configuration buscar la opción Run a getty (login prompt) after boot: en TTY port colocar ttyS0 y en Baudrate seleccionar 115200.

   Buscar la opción Root filesystem overlay directories y poner la ruta de overlay. Ejemplo:

```
/home/milton/Documentos/Embebidos/MinitallerTFLite/
Minitaller_TensorFlowLite/overlay
```

   Volver al menú principal y dirigirse a Filesystem images seleccionar la opción ext2/3/4 root y marcar ext2 = [*] o equivalente. Dirigirse a Exact size y colocar 256M.

   Guardar y salir.

   Ejecuta la construcción con el comando.

```
make -j"$(nproc)"
```

8. Paso. Abrir la imagen y ejecutar.
   Ejecuta el siguiente comando para abrir la imagen una vez se complete de construir

```
cd output/images
qemu-system-x86_64 \
  -m 1024 \
  -kernel bzImage \
  -drive file=rootfs.ext2,if=virtio,format=raw \
  -append "root=/dev/vda console=ttyS0" \
  -serial mon:stdio -nographic
```

Posteriormente aparecerá **buildroot login** colocar root e instalar los paquetes precompilados.

```
python3 -m pip install --no-index --find-links /opt/wheels
numpy==1.26.4 pillow==10.4.0
```

```
python3 -m pip install --no-index --find-links /opt/wheels
/opt/wheels/tflite_runtime-2.14.0-*.whl
```

Los comandos anteriores tienen salto de línea, eliminar antes de ejecutar. Luego ingresa a la ruta y ejecutar el código con lo siguiente:

```
cd /opt/app
python3 classify.py
```

Haz una captura del resultado y documenta en resultados, en la descripción coloca **Resultados de la ejecución en qemu**

Resultados

Figura 1: Golden Retriever.

```
milton@milton-Nitro-AN515-55:~/Documentos/Embebidos/MinitallerTFLite/Minitaller_
TensorFlowLite$ python3 classify.py
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
1. [208] golden retriever: 0.959487
2. [221] Sussex spaniel: 0.008034
3. [264] Pembroke: 0.005361
4. [223] kuvasz: 0.003113
5. [232] collie: 0.001832
milton@milton-Nitro-AN515-55:~/Documentos/Embebidos/MinitallerTFLite/Minitaller_
TensorFlowLite$
```

Figura 2: Resultado de la ejecución en mi computador.

```
Installing collected packages: pillow, numpy
  Attempting uninstall: pillow
    Found existing installation: Pillow 10.0.1
    Uninstalling Pillow-10.0.1:
      Successfully uninstalled Pillow-10.0.1
Successfully installed numpy-1.26.4 pillow-10.4.0
WARNING: Running pip as the 'root' user can result in broken permissions and cov
# python3 -m pip install --no-index --find-links /opt/wheels /opt/wheels/tflite_
runtime-2.14.0-*.whl
Looking in links: /opt/wheels
Processing /opt/wheels/tflite_runtime-2.14.0-cp311-cp311-manylinux2014_x86_64.wl
Requirement already satisfied: numpy>=1.23.2 in /usr/lib/python3.11/site-packag)
Installing collected packages: tflite-runtime
Successfully installed tflite-runtime-2.14.0
WARNING: Running pip as the 'root' user can result in broken permissions and cov
# cd /opt/app
# python3 classify.py
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
1. [208] golden retriever: 0.959487
2. [221] Sussex spaniel: 0.008034
3. [264] Pembroke: 0.005361
4. [223] kuvasz: 0.003113
5. [232] collie: 0.001832
#
```

Figura 3: Resultado de la ejecución en qemu.