

COL788 Embedded Systems

Part 1 Report

Detect car horns on Sensortile M4 micro-controller at high accuracy and low latency (not missing audio)

Pratik Nimbalkar (2020CS10607)

Prakul Virdi (2019CS50442)

Nikhil Unavekar (2020CS10363)

To do

- (a) Train a model including the class car_horn
- (b) Evaluate quantized model accuracy on laptop model
- (c) Evaluate deployed model latency on Sensortile M4

Our Work

1. Built the Project
2. Quantized the model - made some changes to the model
3. Deployed on board

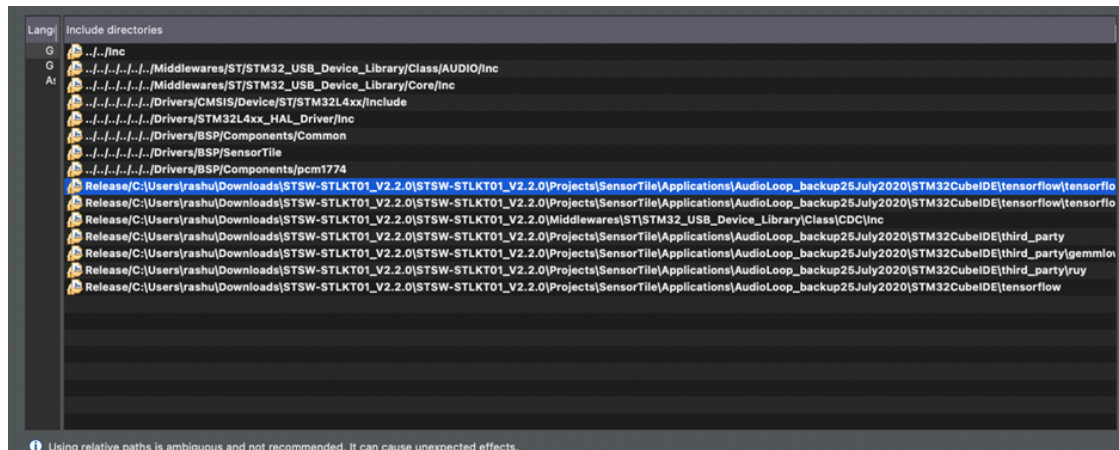
1. Building the Project

1. Used the Optimized Transformer Code folder provided
2. In stm32cubeIDE, Imported this project
Optimized_Transformer_code/STM32CubeIDE/STM32L4xx-SensorTile
3. Once imported, build the project
4. Modified main.cpp. Used Optimized_Transformer_code/KWT_TF2 folder/ "to_cc.py" script to convert model to C++ code in the kwt.cc file.
5. Copied the contents of the kwt.cc file into main.cpp. Replaced weights - "const unsigned char kwt[] DATA_ALIGN_ATTRIBUTE = {}"
7. Built project after changing paths and making some other changes.

I) Changing paths

Changed following mentioned (rashu only) paths according to our system-

C paths -



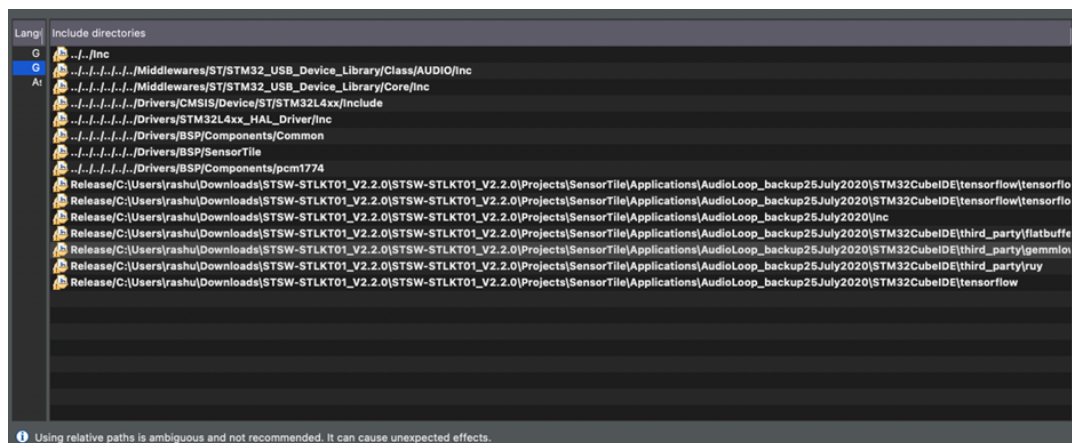
Paths which are not completely visible here -

-
Release/C:\Users\rashu\Downloads\STSW-STLKT01_V2.2.0\STSW-STLKT01_V2.2.0\Projects\SensorTile\Applications\AudioLoop_backup25July2020\STM32CubeIDE\tensorflow\tensorflow\lite\micro\tools\make\downloads\CMSIS\CMSIS\Core\Include

-
Release/C:\Users\rashu\Downloads\STSW-STLKT01_V2.2.0\STSW-STLKT01_V2.2.0\Projects\SensorTile\Applications\AudioLoop_backup25July2020\STM32CubeIDE\tensorflow\tensorflow\lite\micro\tools\make\downloads

-
Release/C:\Users\rashu\Downloads\STSW-STLKT01_V2.2.0\STSW-STLKT01_V2.2.0\Projects\SensorTile\Applications\AudioLoop_backup25July2020\STM32CubeIDE\third_party\gemmlowp

C++ paths -



Paths which are not completely visible here -

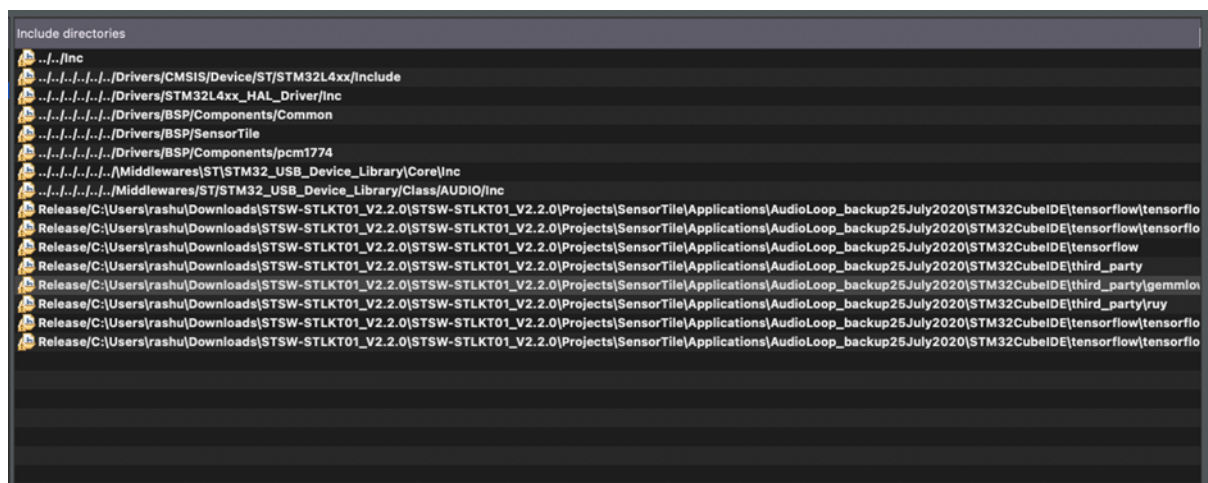
Release/C:\Users\rashu\Downloads\STSW-STLKT01_V2.2.0\STSW-STLKT01_V2.2.0\Projects\SensorTile\Applications\AudioLoop_backup25July2020\STM32CubeIDE\tensorflow\tensorflow\lite\micro\tools\make\downloads\CMSIS\CMSIS\Core\Include

Release/C:\Users\rashu\Downloads\STSW-STLKT01_V2.2.0\STSW-STLKT01_V2.2.0\Projects\SensorTile\Applications\AudioLoop_backup25July2020\STM32CubeIDE\tensorflow\tensorflow\lite\micro\tools\make\downloads

Release/C:\Users\rashu\Downloads\STSW-STLKT01_V2.2.0\STSW-STLKT01_V2.2.0\Projects\SensorTile\Applications\AudioLoop_backup25July2020\STM32CubeIDE\third_party\flatbuffers\include

Release/C:\Users\rashu\Downloads\STSW-STLKT01_V2.2.0\STSW-STLKT01_V2.2.0\Projects\SensorTile\Applications\AudioLoop_backup25July2020\STM32CubeIDE\third_party\gemmlowp

Assembly Paths -



Paths which are not completely visible here -

Release/C:\Users\rashu\Downloads\STSW-STLKT01_V2.2.0\STSW-STLKT01_V2.2.0\Projects\SensorTile\Applications\AudioLoop_backup25July2020\STM32CubeIDE\tensorflow\tensorflow\lite\micro\tools\make\downloads

Release/C:\Users\rashu\Downloads\STSW-STLKT01_V2.2.0\STSW-STLKT01_V2.2.0\Projects\SensorTile\Applications\AudioLoop_backup25July2020\STM32CubeIDE\tensorflow\tensorflow\lite\micro\tools\make\downloads\CMSIS\CMSIS\Core\Include

Release/C:\Users\rashu\Downloads\STSW-STLKT01_V2.2.0\STSW-STLKT01_V2.2.0\Projects\SensorTile\Applications\AudioLoop_backup25July2020\STM32CubeIDE\third_party\gemmlowp

Release/C:\Users\rashu\Downloads\STSW-STLKT01_V2.2.0\STSW-STLKT01_V2.2.0\Projects\SensorTile\Applications\AudioLoop_backup25July2020\STM32CubeIDE\tensorflow\tensorflow\lite\micro\tools\make\downloads\CMSIS\CMSIS\NN\Include

Release/C:\Users\rashu\Downloads\STSW-STLKT01_V2.2.0\STSW-STLKT01_V2.2.0\Projects\SensorTile\Applications\AudioLoop_backup25July2020\STM32CubeIDE\tensorflow\tensorflow\lite\micro\tools\make\downloads\CMSIS\CMSIS\NN\Include

orTile\Applications\AudioLoop_backup25July2020\STM32CubeIDE\tensorflow\tensorflow\lite\micro\tools\make\downloads\cmsis\CMSIS\DSP\Include

Then, we made some other changes like removing the Cortex file (which is unnecessary and was causing some errors) and fixed other dependencies, errors which were present. In usbd_cdc.h file, we removed rashtu dependencies which were earlier causing some error.

This led us to finally building our project successfully -

```
Finished building target: STSW-STLKT01-AudioLoop.elf

arm-none-eabi-size  STSW-STLKT01-AudioLoop.elf
   text    data    bss     dec     hex filename
 650528    504   113480   764512   baa60 STSW-STLKT01-AudioLoop.elf
Finished building: default.size.stdout

arm-none-eabi-objdump -h -S STSW-STLKT01-AudioLoop.elf > "STSW-STLKT01-AudioLoop.list"
Finished building: STSW-STLKT01-AudioLoop.list

arm-none-eabi-objcopy -O binary "STSW-STLKT01-AudioLoop.elf" "STSW-STLKT01-AudioLoop.bin"

22:13:36 Build Finished. 0 errors, 25 warnings. (took 1m:41s.835ms)
```

2. Quantized the model

I) Training model

To train the model, followed these steps -

https://github.com/STMicroelectronics/stm32ai-modelzoo/tree/main/audio_event_detection/scripts/training

Used custom model available at -

https://github.com/STMicroelectronics/stm32ai-modelzoo/tree/main/audio_event_detection/scripts/utis/models

II) To quantize the model -

https://github.com/STMicroelectronics/stm32ai-modelzoo/tree/main/audio_event_detection/scripts/evaluate

III) To deploy the model -

https://github.com/STMicroelectronics/stm32ai-modelzoo/tree/main/audio_event_detection/scripts/deployment

We used the custom model since generated yamnet model size was coming high (beyond maximum limit allowed on sensortile box).

In custom model, we removed the bottom two layers out of three neural layers present as shown -

Removed layers from custom model -

```
22     # Here you can define your own model feature extraction layers
23     # -----
24     x = layers.Conv2D(16, (3, 3), strides=(1, 1), padding='same', use_bias=False)(inputs)
25     x = layers.BatchNormalization()(x)
26     x = layers.Activation('relu')(x)
27     x = layers.MaxPooling2D()(x)
28     x = layers.Conv2D(32, (3, 3), strides=(1, 1), padding='same', use_bias=False)(x)
29     x = layers.BatchNormalization()(x)
30     x = layers.Activation('relu')(x)
31     x = layers.MaxPooling2D()(x)
32     x = layers.Conv2D(64, (3, 3), strides=(2, 2), padding='same', use_bias=False)(x)
33     x = layers.BatchNormalization()(x)
34     x = layers.Activation('relu')(x)
35     x = layers.MaxPooling2D()(x)
36     # -----
```

Removed line 28 to line 35 from this code snippet.

By removing these two layers, we were able to make a smaller size model.

Changed input shape -

Also, we changed the input shape to [32,32] instead of the default value [96,64], i.e. we changed the number of mel spectrogram coefficients from 64 to 32 and the no of samples per window size from 96 to 32 as shown in below figure-

This also helped in reducing the size of the model.

```
feature_extraction:
  patch_length: 32 # 96
  n_mels: 32 # 64
  overlap: 0.25
  n_fft: 512
  hop_length: 160
  window_length: 400
  window: hann
```

Therefore, the input shape changed from [96, 64] to [32,32] as shown -

```
model:
  model_type: {name : custom, embedding_size: 256}
  # Shape of a single patch.
  # Input shape must be [mels, frames]
  input_shape: [32, 32]
  expand_last_dim: True
  transfer_learning : False
  fine_tune : False
  dropout: 0
  multi_label : False
```

Also, we Included car_horn in class_names as initially it did not have it

```
dataset:
  name: custom
  class_names: ['dog', 'chainsaw', 'crackling_fire', 'helicopter', 'rain',
                'crying_baby', 'clock_tick', 'sneezing', 'rooster', 'car_horn']
  audio_path: datasets/ESC-50/audio
  csv_path: datasets/ESC-50/meta/esc50.csv
  file_extension: .wav
  validation_split: 0.1
  test_split : 0.2
  test_path:
  use_other_class: False
  n_samples_per_other_class: 2
```

3. Deployed on board

Followed these steps -

https://github.com/STMicroelectronics/stm32ai-modelzoo/tree/main/audio_event_detection/scripts/deployment

Also, for deploying on board, we had to make some changes in main.cpp with respect to the kwt file.

Modified main.cpp - Used Optimized_Transformer_code/KWT_TF2 folder/ "to_cc.py" script to convert model to C++ code in the kwt.cc file.

Copied the contents of the kwt.cc file into main.cpp. Replaced weights - "const unsigned char kwt[] DATA_ALIGN_ATTRIBUTE = {}"

After this, we used the Deploy button in IDE to deploy the model on the given board.