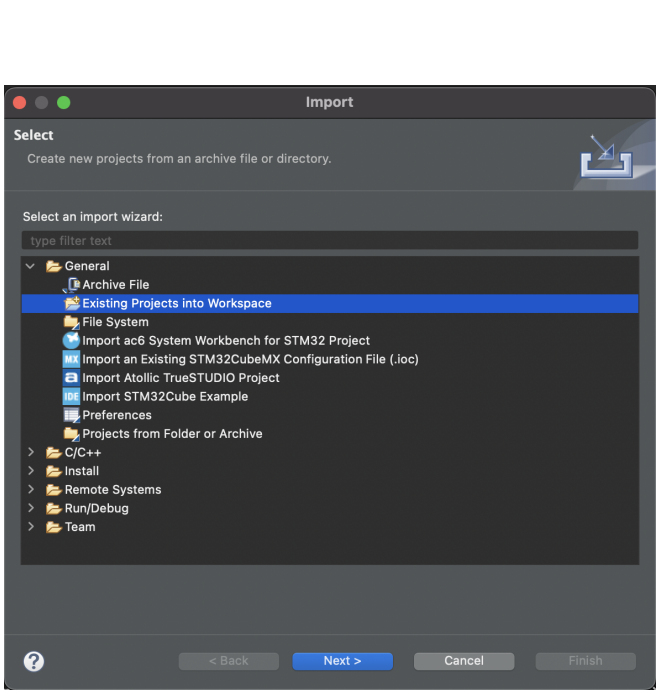
MIT 6.S965 Fall 2022
TinyML and Efficient Deep Learning Computing Course - Lab 4 Instructions

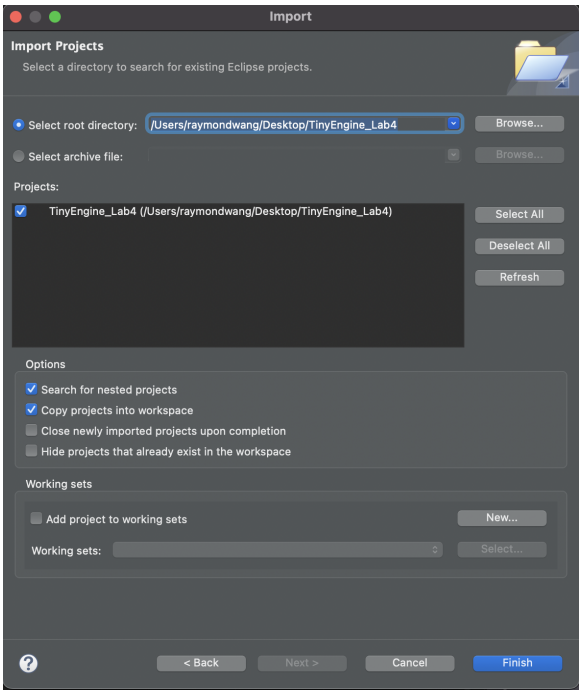
0. If you encounter any problems during deployment, please see the FAQ section at the end of this document.

1. Follow mit-6s965-lab4-tinyml.ipynb on Colab to generate C files

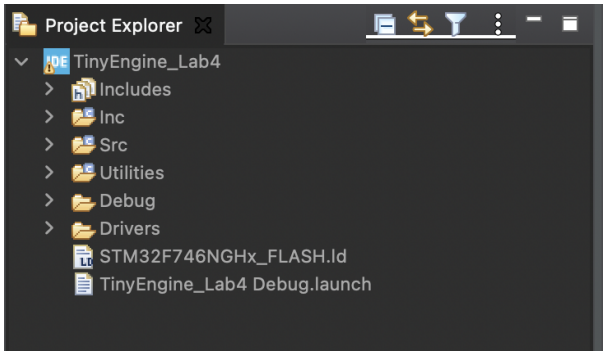
2. Download the codebase of TinyEngine and also STM32CubeIDE, a C/C++ development platform with peripheral configuration, code generation, code compilation, and debug features for STM32 microcontrollers and microprocessors.
   a) Download and install STM32CubeIDE on your computer. Please download **version 1.5.0**.
      - STM32CubeIDE Download Link: https://www.st.com/en/development-tools/stm32cubeide.html#get-software
      - Please refer to "UM2563 STM32CubeIDE installation guide" and "UM2553 STM32CubeIDE quick start guide" for more detailed installation and user guides. Link: https://www.st.com/en/development-tools/stm32cubeide.html#documentation
   b) Download the codebase of the TinyEngine Person Detection Project at: https://hanlab.mit.edu/files/course/labs/TinyEngine_Lab4.zip

3. Set up STM32CubeIDE for compilation and run.
   a) Decompress the zipped file, and get the folder "TinyEngine_Lab4".
   b) Import the project into STM32CubeIDE by: [File] -> [Import…] -> [General] -> [Existing Projects into Workspace] (Import the entire "TinyEngine_Lab4" directory).





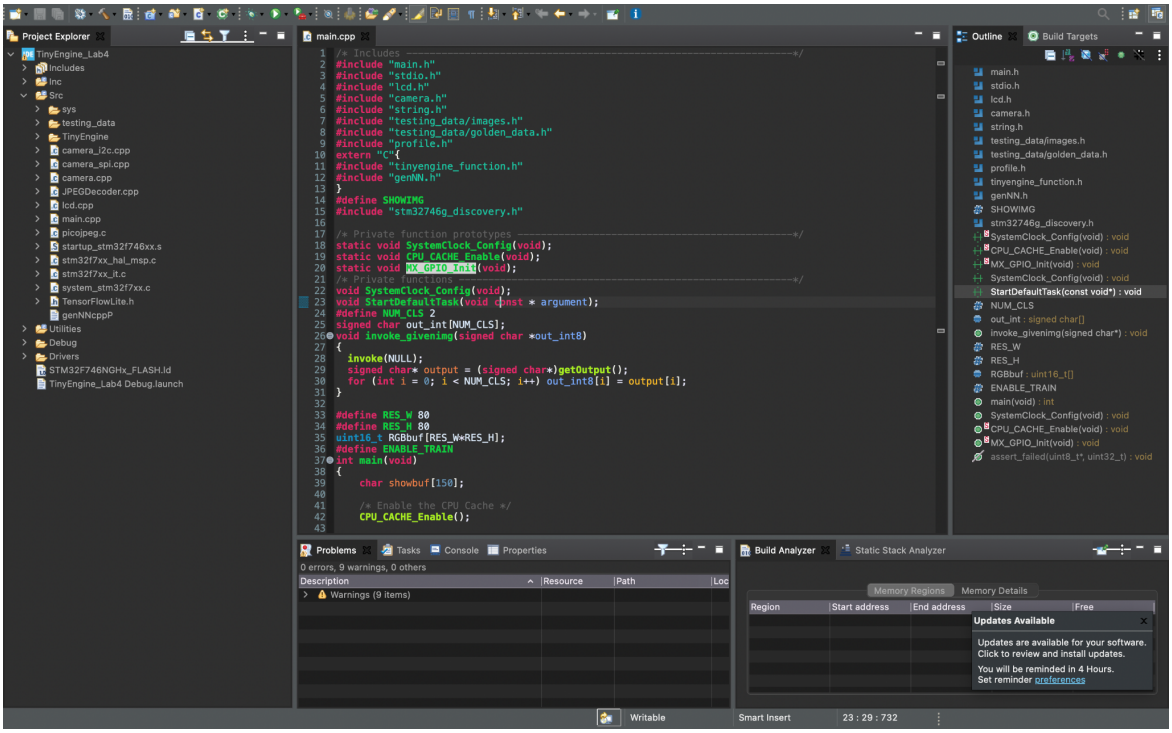
c) After the import, “TinyEngine_Lab4” should be shown in Project Explorer of your STM32CubeIDE as the example figure below:



d) Decompress “codegen.zip” downloaded from Colab, and then copy and paste three C files into your TinyEngine codebase:

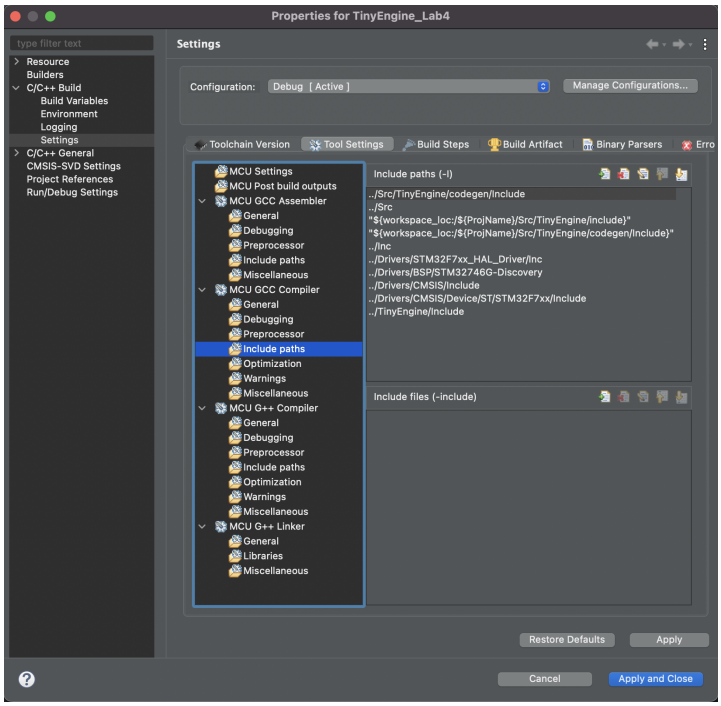
- “codegen/Include/genModel.h” -> “TinyEngine_Lab4/Src/TinyEngine/codegen/Include”
- “codegen/Include/macro.h” -> “TinyEngine_Lab4/Src/TinyEngine/codegen/Include”
- “codegen/Source/genModel.c” -> “TinyEngine_Lab4/Src/TinyEngine/codegen/Source”

e) Open “TinyEngine_Lab4/Src/main.cpp”, as shown in the figure below:

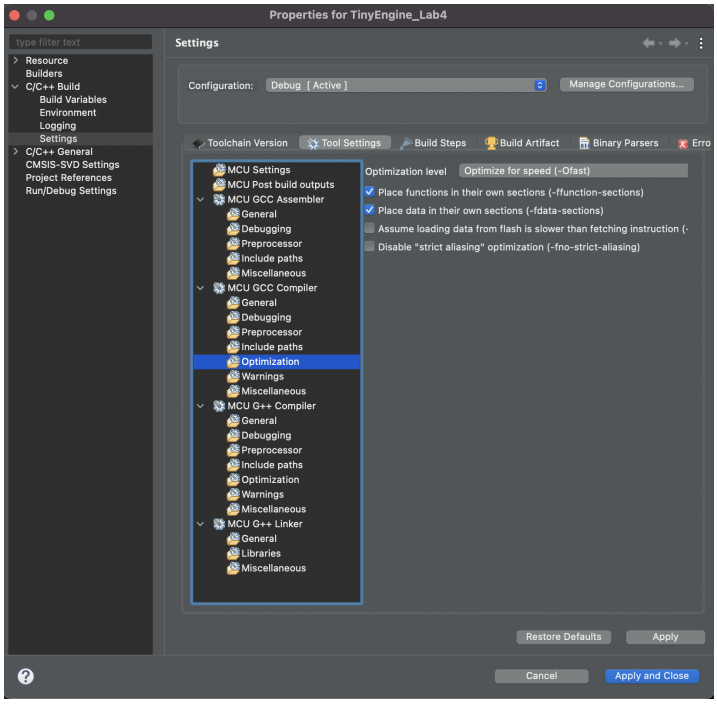


f) Verify you have the correct compilation settings. (The default settings should be correct, but please follow the following steps to ensure that.):

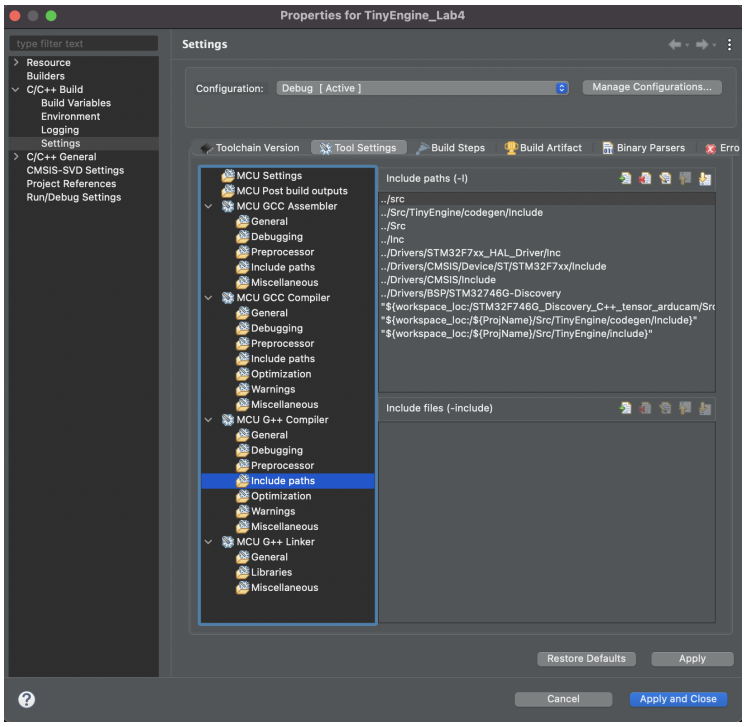
- Set the include paths of GCC compiler by [Project] -> [Properties] -> [C/C++ Build] -> [Settings] -> [Tool Settings] -> [MCU GCC Compiler] -> [Include paths] as the figure below:



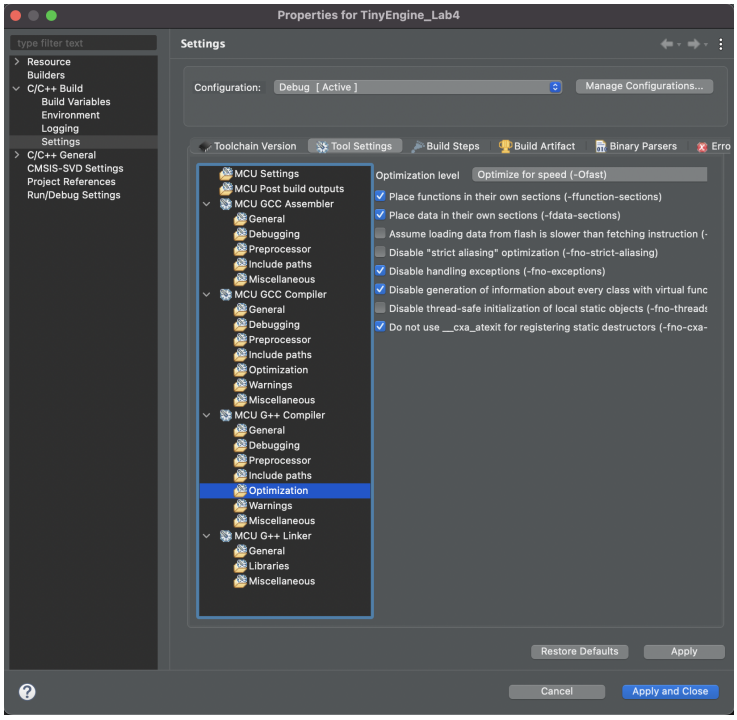
g) Set the optimization level of GCC compiler to “-Ofast” by [Project] -> [Properties] -> [C/C++ Build] -> [Settings] -> [Tool Settings] -> [MCU GCC Compiler] -> [Optimization] as the figure below:



h) Set the include paths of GCC compiler by [Project] -> [Properties] -> [C/C++ Build] -> [Settings] -> [Tool Settings] -> [MCU G++ Compiler] -> [Include paths] as the figure below:

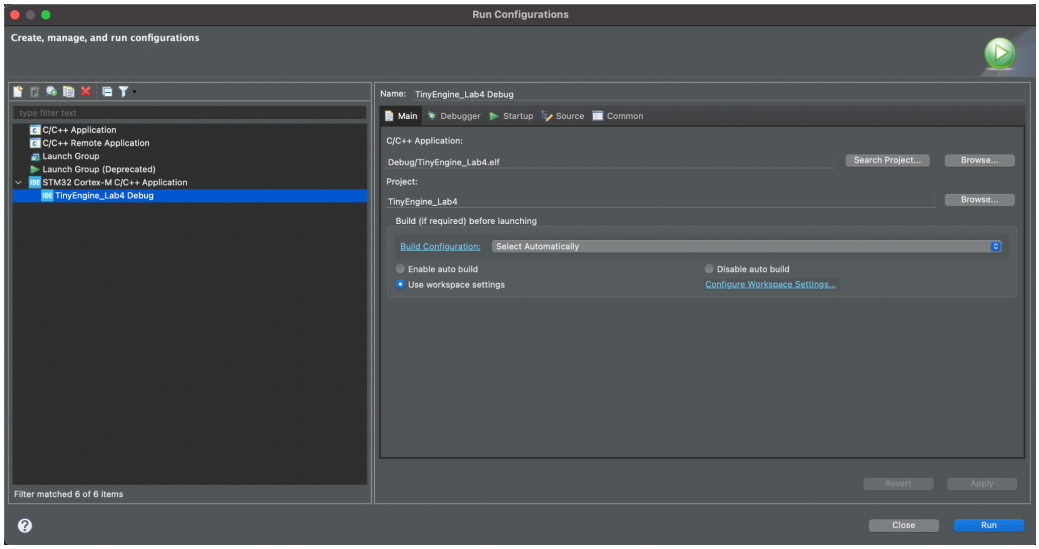


i) Set the optimization level of GCC compiler to “-Ofast” by [Project] -> [Properties] -> [C/C++ Build] -> [Settings] -> [Tool Settings] -> [MCU G++ Compiler] -> [Optimization] as the figure below:

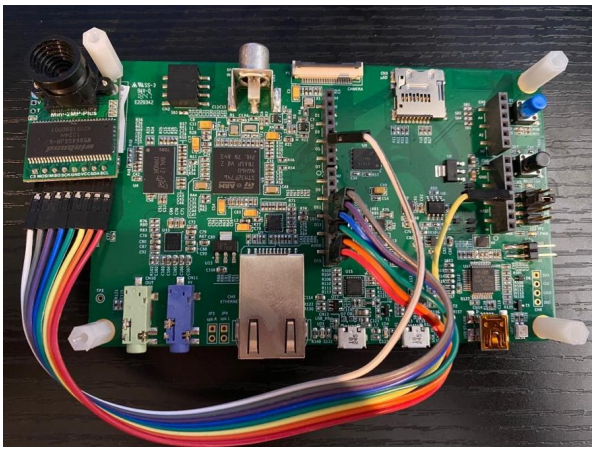


j) Click [Project] -> [Build Project] to build/compile the program and generate the binary executable files.

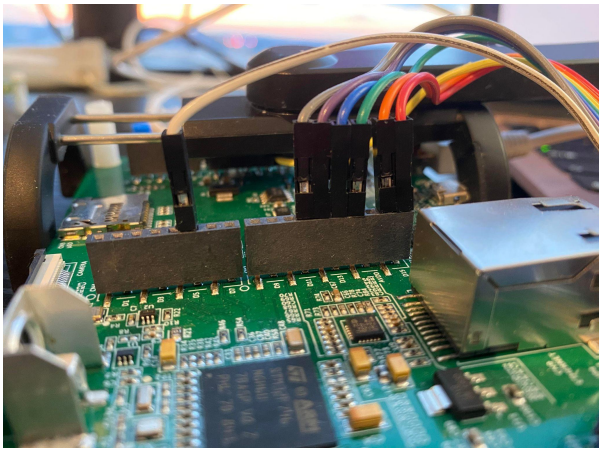
k) Set the run/debug configurations by [Run] -> [Run Configurations…] -> [STM32 Cortex-M C/C++ Application] -> [TinyEngine_Lab4 Debug] -> [C/C++ Application] -> [Browse…]:
- Point to the correct elf file (file path: Debug/TinyEngine_Lab4.elf) to correctly run the program, like the example shown in the figure below:



4. Set up your STM32F746G-DISCO discovery board to connect the Arducam to the board and also establish the USB connection with the board.
   a) Connect your Arducam to the board with jumper wires according to the following PIN connections:
      - SPI: MOSI->PB15(D11), MISO->PB14(D12), SCK->PI_1(D13), CS(NSS)->PI_0(D5), VCC-> 3.3V, GND->GND
      - I2C: SCL->PB8(D15). SDA->PB9(D14)



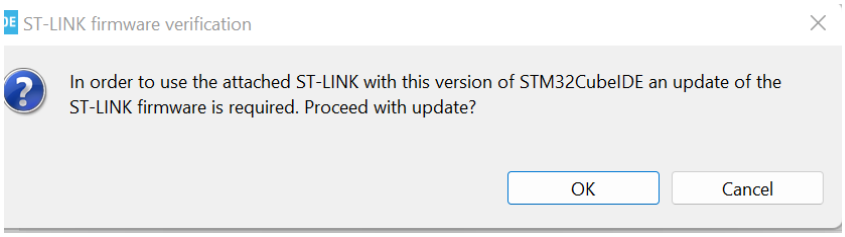
(Top view)



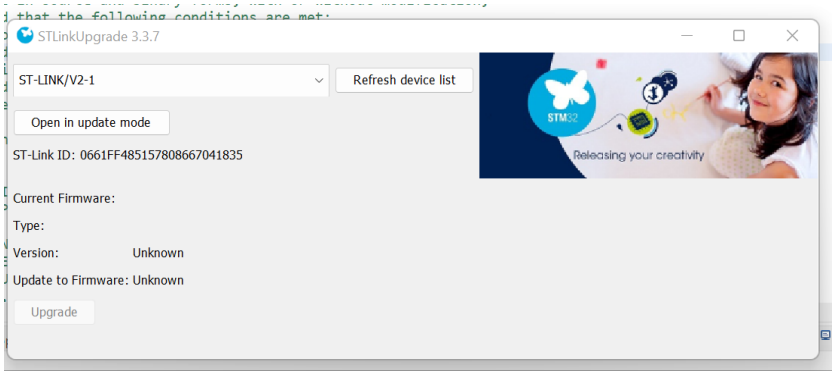
(Side view)

b) Establish the USB connection with the STM32F746G discovery board as shown in the figure below:

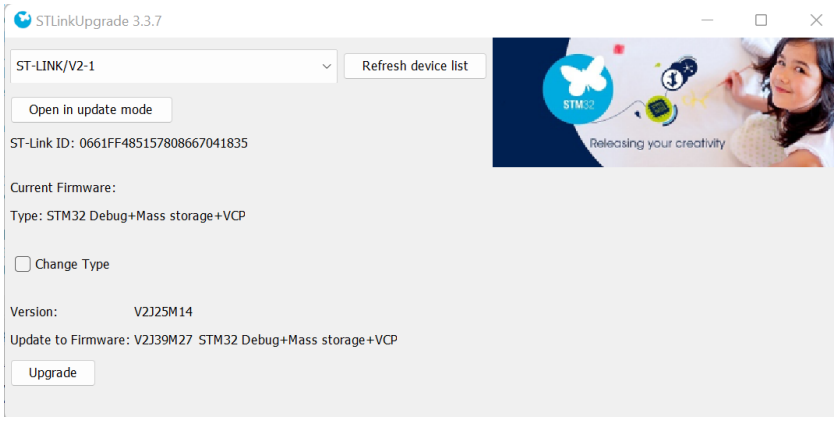
5. Now, let's run the demo!
   a) Click [Run] -> [Run] to execute the binary executable file on your board.
   b) If the system requires updating the ST-LINK firmware, please first click "OK":



   c) Click "Open in update mode":



d) Click “Upgrade”:



e) Click [Run] -> [Run] in the STM32CubeIDE again.

f) If you successfully run the demo, the LCD screen on your STM32F746G-DISCO discovery board should display person detection results (person/no person) and frames per second (FPS). Here is an example demo video: https://www.youtube.com/watch?v=YvioBgtec4U&ab_channel=MITHANLab

6. In the following step, you should iteratively turn on each optimization technique in *optimization_config* of mit-6s965-lab4-tinyml.ipynb on Colab and generate different C source files.
   a) Go back to mit-6s965-lab4-tinyml.ipynb on Colab.
   b) - Set all the optimization switches as *False*, and generate three corresponding C files.
      - Copy and paste the C files into your TinyEngine codebase (Step 3d).
      - Clean the project (right-click [TinyEngine_Lab4] in the project explorer -> [Clean Project]).
      - Rebuild the project (Step 3j).
      - Reset run configurations (Step 3k).
      - Rerun the program (Step 5).
      - Evaluate the inference speed under this optimization setting on your STM32F746G-DISCO discovery board.
   c) Set *loop_reordering* as *True*, and generate three corresponding C files.
      - Copy and paste the C files into your TinyEngine codebase (Step 3d).
      - Clean the project (right-click [TinyEngine_Lab4] in the project explorer -> [Clean Project]).
      - Rebuild the project (Step 3j).
      - Reset run configurations (Step 3k).
      - Rerun the program (Step 5).
      - Evaluate the inference speed under this optimization setting on your STM32F746G-DISCO discovery board.
   d) Set *loop_reordering* and *im2col* as *True*, and generate three corresponding C files.
      - Copy and paste the C files into your TinyEngine codebase (Step 3d).

- Clean the project (right-click [TinyEngine_Lab4] in the project explorer -> [Clean Project]).
- Rebuild the project (Step 3j).
- Reset run configurations (Step 3k).
- Rerun the program (Step 5).
- Evaluate the inference speed under this optimization setting on your STM32F746G-DISCO discovery board.

e) Set *loop_reordering*, *im2col* and *hwc2chw_weight* as *True*, and generate three corresponding C files.
- Copy and paste the C files into your TinyEngine codebase (Step 3d).
- Clean the project (right-click [TinyEngine_Lab4] in the project explorer -> [Clean Project]).
- Rebuild the project (Step 3j).
- Reset run configurations (Step 3k).
- Rerun the program (Step 5).
- Evaluate the inference speed under this optimization setting on your STM32F746G-DISCO discovery board.

f) Set *loop_reordering*, *im2col*, *hwc2chw_weight* and *loop unrolling* as *True*, and generate three corresponding C files.
- Copy and paste the C files into your TinyEngine codebase (Step 3d).
- Clean the project (right-click [TinyEngine_Lab4] in the project explorer -> [Clean Project]).
- Rebuild the project (Step 3j).
- Reset run configurations (Step 3k).
- Rerun the program (Step 5).
- Evaluate the inference speed under this optimization setting on your STM32F746G-DISCO discovery board.

g) Set *loop_reordering*, *im2col*, *hwc2chw_weight*, *loop unrolling* and *simd* as *True*, and generate three corresponding C files.
- Copy and paste the C files into your TinyEngine codebase (Step 3d).
- Clean the project (right-click [TinyEngine_Lab4] in the project explorer -> [Clean Project]).
- Rebuild the project (Step 3j).
- Reset run configurations (Step 3k).
- Rerun the program (Step 5).
- Evaluate the inference speed under this optimization setting on your STM32F746G-DISCO discovery board.

h) Set all the optimization switches (i.e., *loop_reordering*, *im2col*, *hwc2chw_weight*, *loop unrolling*, *simd* and *inplace_depthwise*) as *True*, and generate three corresponding C files.
- Copy and paste the C files into your TinyEngine codebase (Step 3d).
- Clean the project (right-click [TinyEngine_Lab4] in the project explorer -> [Clean Project]).
- Rebuild the project (Step 3j).

- Reset run configurations (Step 3k).
- Rerun the program (Step 5).
- Evaluate the inference speed under this optimization setting on your STM32F746G-DISCO discovery board.

7. Based on your observation results from Step 6, please answer Question 3 on Colab.

8. Please trace the codebase of TinyEngine and answer Question 4 on Colab.

Frequently Asked Questions (FAQ)

1. Why can't I compile the project?
A: Please check whether you download the correct version (1.5.0) of STM32CubeIDE.

2. I successfully compiled the project, but the LCD screen on my MCU board shows nothing but black when running the project.
A: 1) Please check whether you download the correct version (1.5.0) of STM32CubeIDE.
2) Please check your PIN connection (Step 4a).

3. I cannot compile the project, and there are errors after compiling.
A: Please check whether you carefully and completely follow Steps 3 to 6 to run the demo.

4. When running the project, STM32CubeIDE shows error messages like "Can't find gdb server".
A: Please try to restart your computer. Besides, to prevent this bug/error, please remember to click "Eject DIS_F746NG" on your computer every time before you unplug the MCU board.