Joint Track Machine Learning

Generated by Doxygen 1.9.5

1 Compile Instructions for Joint Track Machine Learning **1**

1.1 External Libraries 1

1.1.1 Binary Downloads 1

1.1.2 To be Built 1

1.2 In-House Libraries 2

1.3 External Library CMake Locations 2

1.4 Extras 2

Chapter 1

Compile Instructions for Joint Track Machine Learning

1.1 External Libraries

1.1.1 Binary Downloads

- CUDA Toolkit 11.6 (https://developer.nvidia.com/cuda-11-6-0-download-archive)
  - I usually do the local installer so that I have everything that I need all at once. You will not need to compile this.
- Qt 5.15.2 (https://www.qt.io/download)
  1. "Go Open Source"
  2. Scroll down to "Looking for Qt Binaries" and hit "Download the Qt Online Installer"
  3. You'll want to downlaod the MSVC2019_64 version of Qt. There are some binaries for GCC and other versions of MSVC, but those can get beefy and we don't need them.
  4. Also make sure that you download the newest version of "designer". This gives you an gui framework to edit the JTML gui
- PyTorch "Libtorch" 1.12.1 (https://pytorch.org/)
  - Scroll down and select Stable->Windows->C++/Java->CUDA 11.6

1.1.2 To be Built

MAKE SURE YOU BUILD THE "RELEASE" NOT "DEBUG"

- VTK 7.1.1 (https://vtk.org/download/)
  - You'll download the zip of the source code.
  - You will build this with CMake
  - One thing that you'll need to do in the Visual Studio Project configuration is change "vtkRenderingLabel" to compile with MSVC2017.
  - Once you finish building it, make sure to "build" the "INSTALL" target.
- OpenCV 4.5.5 (https://github.com/opencv/opencv/releases/tag/4.5.5)
  - Download the zip of the source code and build using CMake. It shouldn't be too tough.

1.2 In-House Libraries

These need to be built in order using cmake.

1. CostFunctionTools (https://github.com/ajensen1234/CostFunctionTools-CMake)
    - You won't do any development on this, but you should clone it from github anyway.
    - Build this using CMake, you will need to link some of the previous ^^ external libraries
2. JTA_Cost_Functions (https://github.com/ajensen1234/JTA_Cost_Functions-CMake)
    - You will need to link to CostFunctionTools as well as the external libraries

1.3 External Library CMake Locations

- OpenCv— `path/to/build_dir`
- Pytorch— `/path_to_libtorch/share/cmake/Torch`
- Qt— `C:/Qt/5.15.2/msvc2019_64/lib/cmake/Qt5/`
- CUDA— CMake should find this automatically
- VTK– CMake should find this automatically.

1.4 Extras

You *might* need to download cuDNN separately (https://developer.nvidia.com/cudnn)

- Let me know if this doesn't work and I can send you the binaries and include files directly.