Typical directory structure of an experimental pack shared for Artifact Evaluation


```
Initialization, customization
          ↓
Algorithm , model, training set
          ↓
       Program
          ↓
       Compiler
          ↓
  Binary and libraries

State of the system    Data set
          ↓               ↓
   Run-time environment   ← Simulator
          ↓
      Results          Hardware
          ↓
      Analysis
```
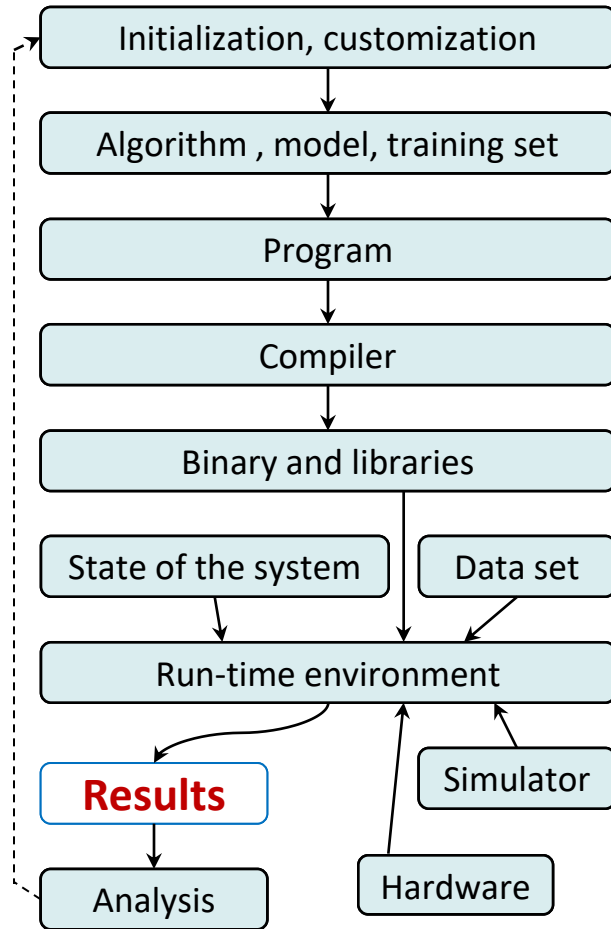
scripts/ download_dataset.sh
download_dnn_model.bat
init_setup_rpi3_gcc7.1.0.sh
init_setup_windows10.bat
init_setup_android.sh
list_programs.sh
compile_program.py
run_program.py
analyze_results.sh
build_predictive_model.bat
plot_graph.sh

programs/ bzip2
classify-image
decode-video-stream

datasets/ jpg-images/ 1.jpg, 2.jpg, 3.jpg
png-images
videos

third-party-tools/ gcc-7.1.0
llvm-4.0
opencl-profiler , cuda-profiler
arch-simulator
caffe, caffe2, tensorflow, cntk, mxnet
clbast, open-blas, viennacl, libdnn

some-meta/ gcc-7.1.0-compiler-flags.txt
llvm-4.0-compiler-flags.txt
rpi3-hw-description.txt

some-results/ reference-speedups.txt
predictions.cvs
graph-autotuning-rpi3.xls

Unified experimental pack in the CK format

.ckr.json *- CK repo name, UID and deps on other CK repos*

module/ **program** / module.py – *unified CK JSON API (functions: compile, run, autotune)*

  dataset
  package
  result
  jnotebook

.cm/ *- UIDs for each CK module*

program/ bzip2
bzip2/.cm/meta.json *- JSON meta for all CK entries*
bzip2/ *.c – program sources*

  classify-image
  decode-video-stream

  .cm *- UIDs for each CK entry (similar to DOI)*

dataset/ image-jpeg-0001
video-frame-0001

package/ compiler-gcc-7.1.0
compiler-llvm-4.0
tool-dvdt-proof
lib-caffe-master-cpu
lib-tensorflow-master-opencl

result/ cgo2017-paper
zlib-autotuning-rpi3

jnotebook/ cgo2017-workflow
cgo2017-graph
rpi3-gcc-autotuning