synaptics

SyNAP Release Notes

Synaptics Confidential. Disclosed Only Under NDA - Limited Distribution.

PN: 511-000xxx-01 Rev 3.0.0

Contents

List of Figures

List of Tables

1. Version 3.0.0

Release date: 2024.03.01

1.1. Runtime

Component	Type	Description
lib	Add	Support for models in .synap format C++ API fully backward compatible with SyNAP 2.x
lib	Add	Support for heterogeneous model execution (NPU, CPU, GPU)
lib	Add	Full support and compatibility with legacy SyNAP 2.x models (*model.nb* and *model.json*)
lib	Add	Integrate Onnx runtime 1.16.3
lib	Add	Integrate TFlite runtime 2.15
lib	Add	TimVx delegate for TFlite for improved online inference on NPU
bin	Add	Optimized model benchmark binary `benchmark_model` integrating TimVx delegate

1.2. Toolkit

Type	Description
Add	Heterogeneous compilation support (NPU, CPU, GPU). The desired delegate(s) can be selected at model compilation time.
Add	Generates *.synap* format by default. The *.synap* format is a bundle that contains both the model subgraph(s) and the companion meta-information. This replaces the *model.nb* and *model.json* files in SyNAP 2.x. It's still possible to generate the legacy *model.nb* and *model.json* files compatible with SyNAP 2.x runtime by specifying the `--out-format nb` option when converting the model
Add	New preprocessing option to accept model input in 32-bits floating point
Fix	Preprocessing support for non-quantized models
Fix	Mixed quantization for tflite models ("No layers to requantize" error)
Fix	Accuracy issues with some models when using mixed quantization
Improve	Inference time for some models with mixed quantization

2. Version 2.8.1

2.1. Runtime

No changes.

2.2. Toolkit

Type	Description
Fix	Import of Tensorflow .nb models
Fix	Import of ONNX models containing MaxPool layers
Fix	Import of ONNX models containing Slice layers

3. Version 2.8.0

3.1. Runtime

Component	Type	Description
lib	Improve	Image Preprocessor now adds horizontal or vertical gray bars when importing an image to preserve proportions. The image is always kept at the center.
lib	Add	Detector now supports 'yolov8' format
lib	Add	Tensor now supports assignent from float and int16 data
driver	Fix	Layer-by-layer profiling now provides more accurate timings
driver	Update	Verisilicon software stack upgraded to Ovxlib 1.1.84
doc	Update	Improvements and clarifications in SyNAP.pdf user manual

3.2. Toolkit

Type	Description
Update	Verisilicon Acuity 6.21.2
Update	Conversion docker updated to Tensorflow 2.13.0 and onnx==1.14.0
Fix	Issues with mixed quantization with some models

4. Version 2.7.0

4.1. Runtime

Component	Type	Description
lib	Add	Face recognition
lib	Add	Optional OpenCV support
lib	Fix	Bounding box scaling in postprocessing for 'yolov5' format
driver	Improve	Load network directly from a user buffer (avoid data copy)
driver	Update	Verisilicon software stack upgraded to Unify driver 6.4.13 and ovxlib 1.1.69
doc	Update	Improvements and clarifications in SyNAP.pdf user manual
doc	Add	Model import tutorial: SyNAP_ModelImport.prf

4.2. Toolkit

Type	Description
Add	Model preprocessing now supports nv12 format
Update	Verisilicon Acuity 6.15.0
Update	Conversion docker to ubuntu 22.04 and tensorflow 2.10.0
Fix	Import of .pb models when post-processing enabled (skip reordering)
Fix	Support relative model pathnames in model_convert.py

5. Version 2.6.0

5.1. Runtime

Component	Type	Description
lib	fix	Fix Tensor::set_buffer in case the same Buffer is assigned/deassigned multiple times.
lib	Add	Tensor assign() supports data normalization
lib	fix	Fix model json parsing for 16-bits models
lib	Add	Preprocessor supports 16-bits models
lib	Add	Preprocessor supports models with preprocessing and cropping
lib	Add	Preprocessor rescale now preserves the input aspect-ratio by default (a gray band is added on the bottom of the image if needed)
lib	Add	Support for scalar tensors
lib	Add	Detector supports yolov5 output format
lib	Add	Buffer sharing (allows to share the tensor memory between different networks avoiding data copy)
lib	Improve	Support 64bits compilation

5.2. Toolkit

Type	Description
Add	Support compilation of models with embedded preprocessing including: format conversion (eg. YUV to RGB), layout conversion (eg. NCHW to NHWC), normalization and cropping
Add	Support “full” model quantization mode
Add	Mixed quantization: the user can mix 8-bits and 16-bits quantization in the same model by specifying the quantization type for each layer
Improve	Quantization images now rescaled preserving the aspect-ratio of the content

6. Version 2.5.0

6.1. Runtime

Component	Type	Description
NNAPI	Improve	Init time for online inference (release mode)
NNAPI	Add	Support for NNAPI compilation cache
lib	Improve	Error checking on out-of-sequence API calls
lib	Add	Move support to Network objects
driver	Fix	Layer-by-layer metrics was not working on some models (inference fail)
driver	Improve	Accuracy of layer-by-layer metrics
driver	Improve	Unify all logcat messages with “SyNAP” tag
driver	Improve	Memory optimization: on-demand loading of compressed OpenVX kernels (saves more than 80MB or RAM)
driver	Change	unified libovxlib.so supporting both VS640 and VS680
driver	Update	Verisilicon software stack upgraded to Unify driver 6.4.11 and ovxlib 1.1.50
driver	Improve	Overall improvements now achieve a score of 33.8 with AIBenchmark 4.0.4

6.2. Toolkit

Type	Description
Update	Verisilicon Acuity 6.9
Add	Support compilation of Caffe models
Improve	Error reporting for quantization issues

7. Version 2.4.0

7.1. Runtime

Component	Type	Description
NNAPI	Fix	Correctly support multiple online models at the same time. Compiling multiple online models in parallel could in some cases give issues (SyNAP HAL crash) in previous releases.
NNAPI	Add	New internal SyNAP model compilation cache. This dramatically improves model initialization time during the first inference. Typical speedup of the first inference is by a factor of 3, can be a factor of 20 or more on some models.
NNAPI	Improve	Further runtime optimizations allowing VS680 to achieve a score of 31.5 in ai-benchmark 4.0.4. This places VS680 at the top position of the IOT group: https://ai-benchmark.com/ranking_IoT_detailed.html
lib	Change	SyNAP default log level is now WARNING (instead of no logs)
doc	Update	Operator support table

7.2. Toolkit

Type	Description
Add	New internal SyNAP model compilation cache allows to improve model compilation time. Typical speedup is by a facor of 3, can be a factor of 20 or more on some models.
Fix	Conversion of ONNX models when output layer name(s) specified explicitly in metafile.

8. Version 2.3.0

8.1. Runtime

Component	Type	Description
all	Add	By-layer profiling support. Low-level driver and runtime binaries and libraries now support layer by layer profiling of any network.
lib	Add	Allocator API in synap device and associated SE-Linux rules. This is the default allocator in libsynapnb and the NNAPI is already making use of it. This also enable any user application (native or not) to execute models without root priviledge, including the synap_cli family.
lib	Add	Sample Java support.
lib	Update	Reorganize libraries. We now have the following libraries: • libsynapnb.so: core EBG execution library • libsynap_preprocess.a: pre-processing • libsynap_postprocess.a: post-processing (classification, detection, etc) • libsynap_img.a: image processing utilities • libsynap_utils.a: common utilities • libsynap_app.a: application support utilities
bin	Add	Repeat mode to synap_cli
bin	Add	EBG for profiling generation to synap_cli_nb
NNAPI	Fix	Memory leak when running models

8.2. Toolkit

Type	Description
Add	by-layer profiling
Add	Secure Model Generation for VS640 (VS680 was already supported) Note: This feature requires special agreement with Synaptics in order to be enabled.

9. Version 2.2.1

9.1. Runtime

Component	Type	Description
lib	Fix	Memory leak when dellocating Buffers
NNAPI	Optimize	Memory savings and simplified dependencies NNRT is now using libsynapnb directly to execute an EBG model; VIPBroker dependency was removed from OVXLIB which is now only used as a graph compiler.

10. Version 2.2.0

10.1. Runtime

Component	Type	Description
all	Add	Linux Baseline VSSDK support
lib	Add	`Preprocessor` class with support for image rescaling and conversion
lib	Add	`Classifier` postprocessor
lib	Add	*Detector*` postprocessors with full support for `TFLite_Detection_PostProcess` layer with external anchors
lib	Add	`Label_info` class
lib	Add	`ebg_utils`: new shared library for EBG format manipulation
lib	Fix	NPU lock functionality
lib	Remove	`nnapi_lock()` API, use vendor.NNAPI_SYNAP_DISABLE property instead. This doesn't require any special permission for the application.
bin	Add	`synap_cli_nb`: new program for NBG to EBG conversion
driver	Optimize	Much reduced usage of contiguous memory
NNAPI	Update	VSI OVXLIB to 1.1.37
NNAPI	Update	VSI NNRT/NHAL to 1.3.1
NNAPI	Add	More operators supported
NNAPI	Optimize	Much higher score for some AI-benchmark models (ex: PyNET and U-Net)
NNAPI	Add	Android CTS/VTS pass for both VS680 and VS640

10.2. Toolkit

Type	Description
Fix	Crash when importing one TFLite object-detection models
Add	Full support for TFLite_Detection_PostProcess layerb
Add	Support for ${ANCHOR} and ${FILE:name} variables in tensor format string
Add	Support for ${ENV:name} variables substitution in model yaml metafile
Add	Support for security.yaml file
Update	VSI acuity toolkit to 6.3.1
Update	Improved error checking
Update	Layer name and shape are now optional when doing quantization
Add	Support for single mean value in metafile
Remove	synap_profile tool
Fix	Handling of relative paths

11. Version 2.1.1

11.1. Runtime

Type	Description
Fix	Timeout expiration in online model execution (ai-benchmark 4.0.4 now runs correctly)
Fix	Issues in `sysfs` diagnostic
Change	On android `synap` logs don't go to `stderr` anymore (just to logcat)

11.2. Toolkit

Type	Description
Fix	`sysfs` section in User Manual
Update	Inference timings section in User Manual now includes y-uv models

12. Version 2.1.0

12.1. Runtime

Type	Description
Add	Full support for SyKURE™: Synaptics secure inference technology
Improve	Tensor Buffers for NNAPI and synapnb now allocated in non-contiguous memory by default
Add	Buildable source code for `synap_cli_ip` sample application
Change	Per-target organization of libraries and binaries in the install tree

12.2. Toolkit

Type	Description
Add	Support for NHWC tensors in rescale layer
Fix	Tensor format in json file for converted models
Improve	Reorganize sections in User Manual

13. Version 2.0.1

13.1. Runtime

Type	Description
Improve	Online inference performance
Add	Option to show SyNAP version in synap_cli application
Add	Buildable source code for all SyNAP sample applications and libraries

13.2. Toolkit

Type	Description
Update	Model coversion tool (fixes offline performance drop in some cases)

14. Version 2.0.0

14.1. Runtime

Type	Description
Improve	Inference engine now supports the new EBG (Executable Binary Graph) model format. Compared to previous NBG format, EBG brings several impovements: • Much faster loading time • Better maintenance and stability (10x lighter driver source code) • Pave the way to secure inference NBG models are not supported anymore.

14.2. Toolkit

Type	Description
Update	Model coversion tools now support EBG generation

15. Version 1.5.0

15.1. Runtime

Type	Description
Add	Synap device information and statistics in sysfs

15.2. Toolkit

Type	Description
Update	Conversion toolkit to v. 5.24.5
Improve	Model quantization algorithm
Add	Generate network information file when model is converted
Add	Host tools binaries and libraries in toolkit/bin and toolkit/lib

16. Version 1.4.0

16.1. Runtime

Type	Description
Fix	CTS/VTS now run successfully with NNAPI

16.2. Toolkit

Type	Description
Update	Conversion toolkit to v. 5.24
Add	Model benchmark binary: /vendor/bin/android_arm_benchmark_model
Add	Model test script and specs

17. Version 1.3.0

17.1. Runtime

Type	Description
Change	Update and cleanup object Detector API
Change	synap_cli_od allows to specify model
Add	synap_cli_od source code
Add	Cmake standalone build for synap_cli_ic and synap_cli_od

17.2. Toolkit

Type	Description
Add	Import and conversion of ONNX models

18. Version 1.2.0

18.1. Runtime

Type	Description
Change	Remove private implementation details from Buffer.hpp
Change	Switch memory allocation to dmabuf
Fix	Model pathnames and documentation for object detection
Add	Synap device
Add	OpenVX headers and librairies

18.2. Toolkit

Type	Description
New	Model quantization support

19. Version 1.1.0

19.1. Runtime

Type	Description
New	NNAPI lock support: `Npu::lock_nnapi()`

19.2. Toolkit

Type	Description
New	Model profiling tool: `synap_profile.py`
New	NNAPI benchmarking script: `synap_benchmark_nnapi.sh`

20. Version 1.0.0

Initial Version.

Copyright

Trademarks

Notice

Contact Us

Visit our website at www.synaptics.com to locate the Synaptics office nearest you.