MINI-AI-520

Kneron KL520 NPU
mPCIe MiniCard Module

User's Manual 2nd Ed

Last Updated: April 10, 2020

Copyright Notice

Acknowledgements

Packing List

Before setting up your product, please make sure the following items have been shipped:

Item	Quantity
● MINI-AI-520 M.2 Module	1
● M3 screw	2

If any of these items are missing or damaged, please contact your distributor or sales representative immediately.

About this Document

This User's Manual contains all the essential information, such as detailed descriptions and explanations on the product's hardware and software features (if any), its specifications, dimensions, jumper/connector settings/definitions, and driver installation instructions (if any), to facilitate users in setting up their product.

Users may refer to the product page on AAEON.com for the latest version of this document.

Safety Precautions

Please read the following safety instructions carefully. It is advised that you keep this manual for future references

1. All cautions and warnings on the device should be noted.
2. Make sure the power source matches the power rating of the device.
3. Position the power cord so that people cannot step on it. Do not place anything over the power cord.
4. Always completely disconnect the power before working on the system's hardware.
5. No connections should be made when the system is powered as a sudden rush of power may damage sensitive electronic components.
6. If the device is not to be used for a long time, disconnect it from the power supply to avoid damage by transient over-voltage.
7. Always disconnect this device from any power supply before cleaning.
8. While cleaning, use a damp cloth instead of liquid or spray detergents.
9. Make sure the device is installed near a power outlet and is easily accessible.
10. Keep this device away from humidity.
11. Place the device on a solid surface during installation to prevent falls.
12. Do not cover the openings on the device to ensure optimal heat dissipation.
13. Watch out for high temperatures when the system is running.
14. Do not touch the heat sink or heat spreader when the system is running
15. Never pour any liquid into the openings. This could cause fire or electric shock.
16. As most electronic components are sensitive to static electrical charge, be sure to ground yourself to prevent static charge when installing the internal components. Use a grounding wrist strap and contain all electronic components in any static-shielded containers.

17. If any of the following situations arises, please the contact our service personnel:
    i. Damaged power cord or plug
    ii. Liquid intrusion to the device
    iii. Exposure to moisture
    iv. Device is not working as expected or in a manner as described in this manual
    v. The device is dropped or damaged
    vi. Any obvious signs of damage displayed on the device
18. Do not leave this device in an uncontrolled environment with temperatures beyond the device's permitted storage temperatures (see chapter 1) to prevent damage.
19. Do NOT disassemble the motherboard so as not to damage the system or void your warranty.
20. If the thermal pad had been damaged, please contact AAEON's salesperson to purchase a new one. Do NOT use those of other brands.
21. The Hex Cylinder Coppers on the front panel are not removable.
22. Repeatedly assemble and disassemble the system may cause damages to the exterior paint and surface and screw holes.
23. Use the right size screwdriver.
24. Use the screwdriver correctly to remove screws from the system.

FCC Statement

This device complies with Part 15 FCC Rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received including interference that may cause undesired operation.

Caution:

There is a danger of explosion if the battery is incorrectly replaced. Replace only with the same or equivalent type recommended by the manufacturer. Dispose of used batteries according to the manufacturer's instructions and your local government's recycling or disposal directives.

Attention:

Il y a un risque d'explosion si la batterie est remplacée de façon incorrecte.
Ne la remplacer qu'avec le même modèle ou équivalent recommandé par le constructeur. Recycler les batteries usées en accord avec les instructions du fabricant et les directives gouvernementales de recyclage.

China RoHS Requirements (CN)

产品中有毒有害物质或元素名称及含量

AAEON Embedded Box PC/ Industrial System

部件名称	有毒有害物质或元素					
	铅 (Pb)	汞 (Hg)	镉 (Cd)	六价铬 (Cr(VI))	多溴联苯 (PBB)	多溴二苯醚 (PBDE)
印刷电路板及其电子组件	○	○	○	○	○	○
外部信号连接器及线材	○	○	○	○	○	○
外壳	○	○	○	○	○	○
中央处理器与内存	○	○	○	○	○	○
硬盘	○	○	○	○	○	○
电源	○	○	○	○	○	○

O：表示该有毒有害物质在该部件所有均质材料中的含量均在 SJ/T 11363-2006 标准规定的限量要求以下。

X：表示该有毒有害物质至少在该部件的某一均质材料中的含量超出 SJ/T 11363-2006 标准规定的限量要求。

备注：
一、此产品所标示之环保使用期限，系指在一般正常使用状况下。
二、上述部件物质中央处理器、内存、硬盘、电源为选购品。

China RoHS Requirement (EN)

Poisonous or Hazardous Substances or Elements in Products

AAEON Embedded Box PC/ Industrial System

Component	Poisonous or Hazardous Substances or Elements					
	Lead (Pb)	Mercury (Hg)	Cadmium (Cd)	Hexavalent Chromium (Cr(VI))	Polybrominated Biphenyls (PBB)	Polybrominated Diphenyl Ethers (PBDE)
PCB & Other Components	O	O	O	O	O	O
Wires & Connectors for External Connections	O	O	O	O	O	O
Chassis	O	O	O	O	O	O
CPU & RAM	O	O	O	O	O	O
Hard Disk	O	O	O	O	O	O
PSU	O	O	O	O	O	O

O：The quantity of poisonous or hazardous substances or elements found in each of the component's parts is below the SJ/T 11363-2006-stipulated requirement.

X: The quantity of poisonous or hazardous substances or elements found in at least one of the component's parts is beyond the SJ/T 11363-2006-stipulated requirement.

Note: The Environment Friendly Use Period as labeled on this product is applicable under normal usage only

Table of Contents

Chapter 1

Product Specifications

1.1 MINI-AI-520 Kneron NPU Module Specifications

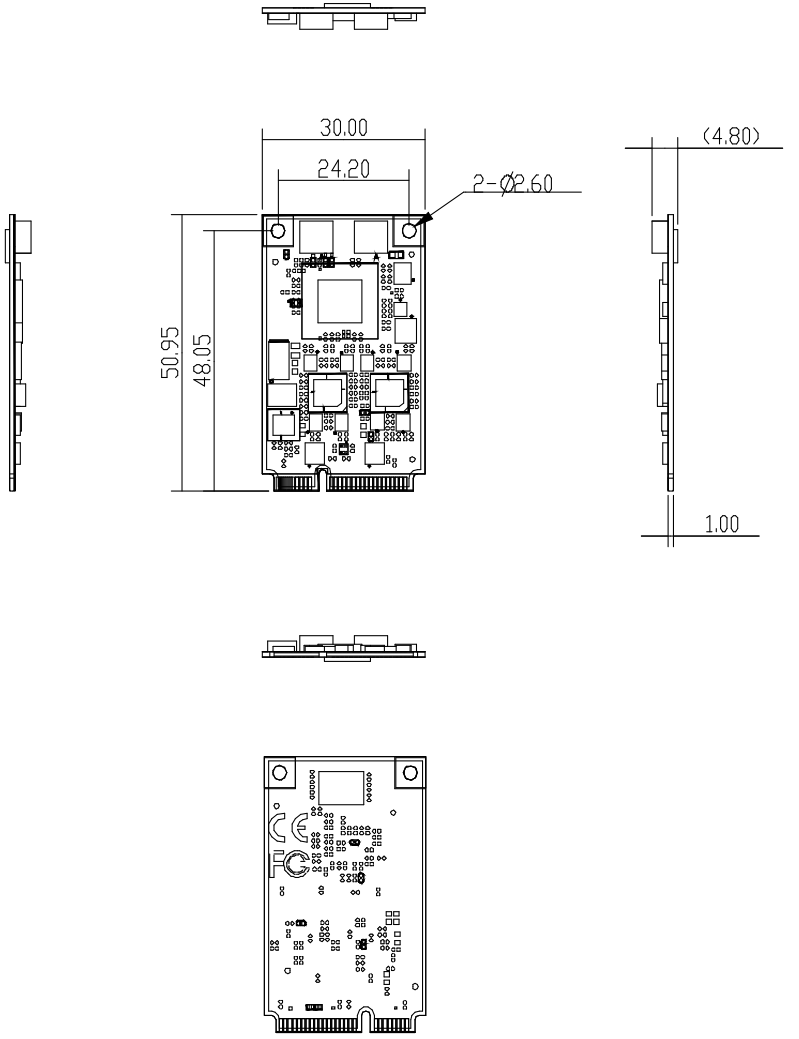
System	
IC	Kneron KL520
Type	Integrated SoC
Support Framework	ONNX, TensorFlow, Keras, Caffe
Support Model	Vgg16, Resnet, GoogleNet, YOLO, Tiny YOLO, Lenet, MobileNet, DenseNet
Memory Type	LPDDR2
NPU Power Efficiency	0.56TOPS/W
Overall Power Consumption	0.5W

Other Specifications	
Operating Temperature	32°F ~ 158°F (0°C ~ 70°C)
Storage Temperature	-40°F ~ 185°F (-40°C ~ 85°C)
Operating Humidity	0% ~ 90% relative humidity, non-condensing
Certification	CE/FCC Class A

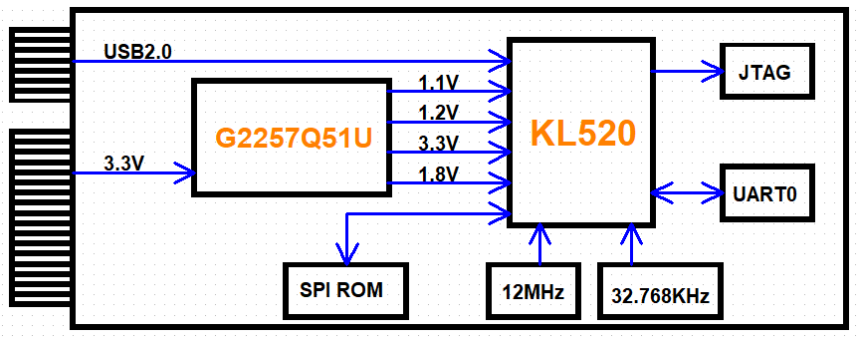
Chapter 2

Hardware Information

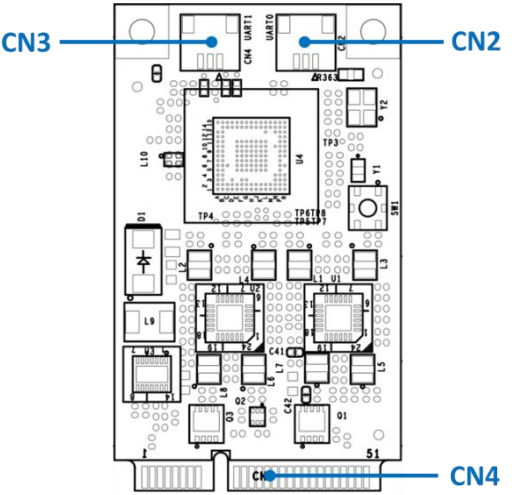
2.1 Dimensions



2.2 Block Diagram



2.3 Board Design

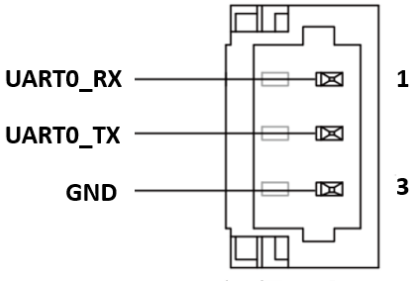


2.4 List of Connectors

This section details the connectors featured on the AI Core X module. This is a reference to help with setup and configuration for your application.

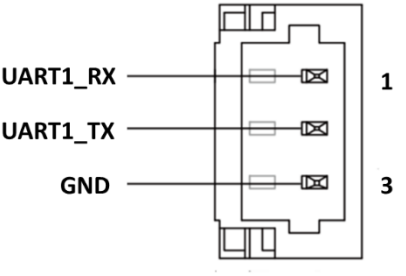
Label	Connector Type
CN2	UART Connector
CN3	UART Connector
CN4	Mini-Card Connector

2.4.1 UART Connector (CN2)



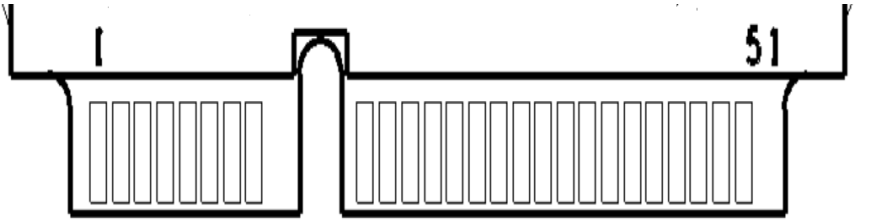
Pin	Signal Description
1	UART0_RX
2	UART0_TX
3	GND

2.4.2 UART Connector (CN3)



Pin	Signal Description
1	UART1_RX
2	UART1_TX
3	GND

2.4.3 Mini-Card Connector (CN4)



Pin	Signal Description	Pin	Signal Description
1	X_PTN	27	GND
2	3.3V_M2	28	NC
3	NC	29	GND
4	GND	30	NC
5	NC	31	NC

Pin	Signal Description	Pin	Signal Description
6	NC	32	NC
7	NC	33	NC
8	NC	34	GND
9	GND	35	GND
10	NC	36	USB_D-
11	NC	37	GND
12	NC	38	USB_D+
13	NC	39	3.3V_M2
14	NC	40	GND
15	GND	41	3.3V_M2
16	NC	42	NC
17	NC	43	GND
18	GND	44	NC
19	NC	45	NC
20	X_PTN	46	NC
21	GND	47	NC
22	NC	48	NC
23	NC	49	NC
24	3.3V_M2	50	GND
25	NC	51	NC
26	GND	52	3.3V_M2

Appendix A

Toolchain User Manual

A.1 Toolchain User Manual

The following attachment, Toolchain User Manual, is supplied with this document and is meant for use with AAEON products featuring the Kneron KL520 NPU Module. If you have any questions regarding this document or your AAEON product, please contact your sales representative for assistance.

Kneron Toolchain User Guide

for AAEON Products with KL520 NPU

Table of Contents

Chapter 1 Overview

KDP toolchain is a software integrating a series of libraries to simulate the operation in the hardware KDP 520. Table 1 shows the list of functions KDP520 supports.

Table 1: KDP520 Supported Functions

Layers/Modules	Functions/Parameters	Spec.
Convolution	Convolution kernel dimentison:	1x1 up to 11x11
	Stride	1,2,4
	Padding:	0-15
	Depthwise Conv	Yes
	Deconvolution	Use Upsampling + Conv
Pooling	Max pooling 3x3	stride 1,2,3
	Max pooling 2x2	stride 1,2
	Ave Pooling 3x3	stride 1,2,3
	Ave Pooling 2x2	stride 1,2
	global ave pooling	Support
	global max pooling	Support
Activation	ReLu	Support
	Leaky ReLU	Support
	PReLU	Support
	ReLU6	Support
Other processing	Batch Normalization	Support
	Add	Support
	Concatenation	Support
	Dense/Fully Connected	Support
	Flatten	Support

Chapter 2 Introduction

2.1 Work Flow

To fully utilize Kneron SDK and get detailed information from the running programs, besides the toolchain GUI, Kneron provides a Linux command toolchain containing the following functions:

(1) Converting deep learning models from different deep learning frameworks (Keras, Tensorflow, Pytorch, Caffe) to ONNX format;

(2) Conducting fixed pointer analysis on the selected model and image dataset; compiling the related model file to Kneron IP's corresponding instructions, weight file, and data flow controls;

(3) Running IP evaluator, as well as simulator and emulator on the selected model.

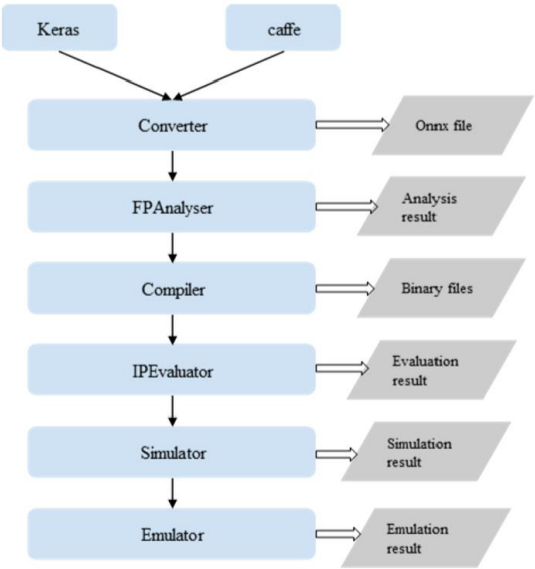


Figure 1. Diagram of working flow

Chapter 3 Docker Installation

3.1 System Requirements

System must be running Ubuntu 16.04

3.2 Installation

Open Terminal and enter the following command:

```
$ sudo chmod +x install_docker
```

Next, enter the following command:

```
$ sudo ./install_docker
```

Docker is now installed.

Chapter 4 Sample Tutorial

4.1 Start the Docker Image

After installing docker, you can start the docker image you just pulled, and get a docker container to run the toolchain. When you start it, you need to configure a local folder as the one for communicating between your local environment and the container. For this example, let's call it Interactive Folder Assume the absolute path of the folder you configure is absolute_path_of_your_folder.

The start command is:

```
docker run -it --rm -v absolute_path_of_your_folder:/data1
kneron/toolchain:linux_command_toolchain
```

For example, if the absolute path of the path folder you configure is /home/aaeon/Document/test_docker, and then the related command is

```
docker run -it --rm -v /home/aaeon/Document/test_docker:/data1
kneron/toolchain:linux_command_toolchain
```

After running the start command, you'll enter into the docker container. Then, copy the example materials to the Interactive Folder by the following command:

```
cp -r /workspace/examples/* /data1/
```

4.2 Converter

4.2.1 Keras to ONNX

Since the Onet model is in Keras format, you need to convert it from Keras to ONNX by the following command:

```
python /workspace/onnx-keras/generate_onnx.py -o
absolute_path_of_output_model_file
absolute_path_of_input_model_file -O -C --duplicate-shared-weights
```

For example:

```
python /workspace/onnx-keras/generate_onnx.py -o /data1/onet-
0.417197.onnx /data1/keras/onet0.417197.hdf5 -O -C --duplicate-
shared-weights
```

There might be some warning log when running this problem, and you can check whether the convert works successfully by checking whether the onnx file is generated.

If there's customized input shape for the model file, you need to use the following command:

```
python /workspace/onnx-keras/generate_onnx.py
absolute_path_of_input_model_file -o
absolute_path_of_output_model_file -I 1 model_input_width
model_input_height num_of_channel
```

4.2.2 Tensorflow to ONNX

Use the following command to convert from Tensorflow to ONNX:

```
/workspace/scripts/tf2onnx.sh absolute_path_of_input_model_file
absolute_path_of_output_onnx_model_file name_of_input_layer:0
name_of_output_layer:0
```

For example:

```
/workspace/scripts/tf2onnx.sh /data1/tensorflow/model/mnist.pb
/data1/mnist.pb.onnx Placeholder:0 fc2/add:0
```

4.2.3 Pytorch to ONNX

Use the following command to convert from Pytorch to ONNX:

```
python /workspace/scripts/pytorch2onnx.py
absolute_path_of_input_model_file channel_number model_input_height
model_input_width absolute_path_of_output_model_file
```

For example:

```
python /workspace/scripts/pytorch2onnx.py
/data1/pytorch/models/resnet34.pth 3 224 224 /data1/resnet34.onnx
```

4.2.4 Pytorch-ONNX to ONNX

Although pytorch support to produce onnx file, it still needs to be converted here.

```
python /workspace/scripts/pytorch2onnx.py
absolute_path_of_input_pytorch_onnx_model_file channel_number
model_input_height model_input_width
absolute_path_of_output_model_file
```

4.2.5 Caffe to ONNX

Use the following command to convert from Caffe to ONNX:

```
python /workspace/onnx-caffe/generate_onnx.py -o
absolute_path_of_output_onnx_model_file -w
absolute_path_of_input_caffe_weight_file -n
absolute_path_of_input_caffe_model_file
```

For example:

```
python /workspace/onnx-caffe/generate_onnx.py -o
/data1/mobilenetv2.onnx -w
/data1/caffe/models/mobilenetv2.caffemodel -n
/data1/caffe/models/mobilenetv2.prototxt
```

4.2.6 ONNX to ONNX

If you have you own onnx file or if there's problem with the onnx converted by the command above, you need to run the following command:

```
python /workspace/scripts/onnx2onnx.py
absolute_path_of_your_input_onnx_model_file -o
absolute_path_of_output_onnx_model_file (-m)
```

Add –m when there is customized layer in your model.

This operation will optimize the mathematical computation in your model.

4.2.7 Edit Function

4.2.7.1 Feature

There is a script called edit.py in the folder /workspace/scripts, and it is an simple ONNX editor which achieves the following functions:

(1) Add nop BN or Conv nodes.

(2) Delete specific nodes or inputs.

(3) Cut the graph from certain node (Delete all the nodes following the node).

(4) Reshape inputs and outputs

4.2.7.2 Usage

Usage of the Edit Function is as follows:

```
editor.py [-h] [-c CUT_NODE [CUT_NODE ...]]
[--cut-type CUT_TYPE [CUT_TYPE ...]]
[-d DELETE_NODE [DELETE_NODE ...]]
[--delete-input DELETE_INPUT [DELETE_INPUT ...]]
[-i INPUT_CHANGE [INPUT_CHANGE ...]]
[-o OUTPUT_CHANGE [OUTPUT_CHANGE ...]]
[--add-conv ADD_CONV [ADD_CONV ...]]
[--add-bn ADD_BN [ADD_BN ...]]
in_file out_file
```

Edit an ONNX model. The processing sequense is 'delete nodes/values' -> 'add nodes' -> 'change shapes'. Cutting cannot be done with other operations together.

Positional arguments:

`in_file`

input ONNX FILE

`out_file`

ouput ONNX FILE

Optional arguments:

`-h, --help`

show this help message and exit

`-c CUT_NODE [CUT_NODE ...], --cut CUT_NODE [CUT_NODE ...]`

remove nodes from the given nodes(inclusive)

`--cut-type CUT_TYPE [CUT_TYPE ...]`

remove nodes by type from the given nodes(inclusive)

`-d DELETE_NODE [DELETE_NODE ...], --delete DELETE_NODE [DELETE_NODE ...]`

delete nodes by names and only those nodes

`--delete-input DELETE_INPUT [DELETE_INPUT ...]`

delete inputs by names

`-i INPUT_CHANGE [INPUT_CHANGE ...], --input INPUT_CHANGE [INPUT_CHANGE ...]`

change input shape (e.g. -i 'input_0 1 3 224 224')

`-o OUTPUT_CHANGE [OUTPUT_CHANGE ...], --output OUTPUT_CHANGE [OUTPUT_CHANGE ...]`

change output shape (e.g. -o 'input_0 1 3 224 224')

`--add-conv ADD_CONV [ADD_CONV ...]`

add nop conv using specific input

`--add-bn ADD_BN [ADD_BN ...]`

add nop bn using specific input

4.2.7.3 Example

(1) In the /workspace/scripts/res folder, there is a VDSR model from Tensorflow. Convert this model first.

```
cd /workspace/scripts && ./tf2onnx.sh res/vdsr_41_20layer_1.pb
res/tmp.onnx images:0 output:0
```

(2) This ONNX file seems valid. But, it's channel last for the input and output. It is using Transpose to convert to channel first, affecting the performance. Use the editor to delete the Transpose and reset the shapes.

```
cd /workspace/scripts && python editor.py res/tmp.onnx new.onnx -d
Conv2D__6 Conv2D_19__84 -i 'images:0 1 3 41 41' -o 'output:0 1 3 41
41'
```

Now, it has no Transpose and takes channel first inputs directly.

4.3 FpAnalyser, Compiler and IpEvaluator

4.3.1 Fill Input Parameters

Before running the programs, you need to configure the input parameters by the *input_params.json* in Interactive Folder. The initial file of *input_params.json* is for Keras Onet model. You can see the detailed explanation for the input parameters in the FAQ Question 1.

4.3.2 Running the Program

After filling the related parameters in input_params.json, you can run the programs by the following command:

```
cd /workspace/scripts && ./fpAnalyserCompilerIpevaluator.sh.x
```

After running this program, the folders called compiler and fpAnalyser will be generated in the Interactive tFolder, which store the result of compiler, ipEvaluator and fpAnalyser.

4.3.3 Get the Result

In Interactive Folder, you'll find a folder called fpAnlayer, which contains the preprocessed image txt files; a folder called compiler, which contains the binary files generated by compiler, as well as evaluation result of ipEvaluator.

4.4 Simulator and Emulator

4.4.1 Fill the Input Parameters

Fill the simulator and emulator input parameters in the *input_params.json* in Interactive Folder. Please refer to the FAQ question 1 to fill the related parameters.

4.4.2 Running the Programs

For running the simulator:

```
cd /workspace/scripts && ./simulator.sh.x
```

And a folder called simulator will be generated in Interactive Folder, which stores the result of the simulator.

For running the emulator:

```
cd /workspace/scripts && ./emulator.sh.x
```

And a folder called emulator will be generated in Interactive Folder, which stores the result of the emulator.

4.4.3 Get the Results

In Interactive Folder, you'll find a folder called simulator, which contains the output files of simulator; a folder called emulator, which contains the output folders of simulator.

In each folder, there are three files: one is the input image file, one whose format is "temp***.txt" is the output of the last layer, and the other one is the preprocess image result.

4.5 Compiler and Evaluator

This part is similar with part 3.4, and the difference is that this part does not run fpAnalyser, it can be used when your model structure is prepared but hasn't been trained.

4.5.1 Fill the Input Parameters

Fill the simulator and emulator input parameters in the input_params.json in Interactive Folder. Please refer to the FAQ question 1 to fill the related parameters.

4.5.2 Running the Programs

For running the compiler and ip evaluator:

```
cd /workspace/scripts && ./compilerIpevaluator.sh.x
```

And a folder called simulator will be generated in Interactive Folder, which stores the result of the compiler and ipEvaluator.

4.5.3 Get the Result

In Interactive Folder, you'll find a folder called compiler, which contains the output files of the compiler and ipEvaluator.

4.6 FpAnalyser and Batch-Compile

This part is the instructions for batch-compile, which will generate the binary file requested by firmware.

4.6.1 Fill the Input Parameters

Fill the simulator and emulator input parameters in the */data1/batch_compile_input_params.json* in Interactive Folder. Please refer to the FAQ question 7 to fill the related parameters.

The following two examples show how to configure the batch_compile_input_params.json.

(1) tiny_yolo_v3

```
{
"input image folder": ["/data1/caffe/images"],
"img_channel": ["RGB"],
"model_input_width": [224], "model_input_height": [224],
"img_preprocess_method": ["yolo"], "input_onnx_file":
["/data1/yolov3-tiny-224.h5.onnx"],
"keep_aspect_ratio": "True",
"command_addr": "0x30000000",
"weight_addr": "0x40000000",
"sram_addr": "0x50000000", "dram_addr": "0x60000000",
"whether_encryption": "No", "encryption_key": "0x12345678",
"model_id_list": [19], "model_version_list": [1]
}
```

(2) tiny_yolo_v3 and Onet

```
{ "input_image_folder": ["/data1/caffe/images",
"/data1/keras/n000645"],
"img_channel": ["RGB", "L"], "model_input_width": [224, 48],
"model_input_height": [224, 48], "img_preprocess_method": ["yolo",
"kneron"],
"input_onnx_file": ["/data1/yolov3-tiny-224.h5.onnx", "/data1/onet-
0.417197.onnx"],
"keep_aspect_ratio": "True",
"command addr": "0x30000000",
"weight_addr": "0x40000000", "sram_addr": "0x50000000",
"dram_addr": "0x60000000",
"whether_encryption": "No",
"encryption_key": "0x12345678", "model_id_list": [19, 20],
"model_version_list": [1, 1] }
```

4.6.2 Running the Programs

For running the compiler and ip evaluator:

```
cd /workspace/scripts && ./fpAnalyserBatchCompile.sh.x
```

And a folder called batch_compile will be generated in Interactive Folder, which stores the result of the fpAnalyer and batch-compile. (all_models.bin & fw_info.bin are in batch compile/compiler folder)

4.6.3 Get the Result

In Interactive Folder, you'll find a folder called compiler, which contains the output files of the fpAnalyer and batch-compile. If you have questions for the meaning of the output files, please refer to the FAQ question 8.

4.7 Draw YOLO Result on Images

The toolchain also provides the function of drawing final result on images for yolo model, i.e. drawing the box and class name.

4.7.1 Steps

(1) Follow the part 3.4 Simulator and Emulator, it will generate the result of emulator for multiply images, and the result folder path is "/data1/emulator". In this folder, the original image, the preprocess image txt file and the final output of the model are classified in different folder.

(2) run the scripts to draw the yolo result

```
cd /workspace/scripts/utils/yolo && python
convert_sim_result_yolo.py
```

After this step, the drawing result will be saved in the subfolders of "/data1/emulator", with the format "imgname_thresh_xxx.png", xxx means the threshold for the box score, which means only the boxes with score higher than this threshold will be drawn in this image.

4.8 FAQ

4.8.1 How to configure the input_params.json?

By following the above instructions, the input_params.json will be saved in Interactive Folder.

Please do not change the parameters' names.

The parameters in input_params.json are:

(1) input_image_folder

The absolute path of input image folder for fpAnalyser.

(2) img_channel

Options: L, RGB

The channel information after the input image is preprocessed. L means single channel. Input for fpAnalyer.

(3) model_input_width

The width of the model input size.Input for fpAnalyer.

(4) model_input_height

The height of the model input size.

(5) img_preprocess_method

Options: kneron, tensorflow, yolo, caffe, pytorch

The image preprocess methods, input for fpAnlayer, and the related formats are following:

"kneron": RGB/256 - 0.5,

"tensorflow": RGB/127.5 - 1.0,

"yolo": RGB/255.0

"pytorch": (RGB/255. -[0.485, 0.456, 0.406]) / [0.229, 0.224, 0.225]

"caffe"(BGR format) BGR - [103.939, 116.779, 123.68]

"customized": please refer to FAQ question 9

(6) input_onnx_file

The absolute path of the onnx file, which works as the input file for fpAnalyser.

(7) keep_aspect_ratio

Options: True, False

Indicates whether or not to keep the aspect ratio.

(8) command_addr

Address for command, input for compiler.

(9) weight_addr

Address for weight, input for compiler.

(10) sram_addr

Address for sram, input for compiler.

(11) dram_addr

Address for dram, input for compiler.

(12) whether_encryption

Option: Yes, No

Whether add encryption on the bin files generated by compiler, input for compiler.

(13) encryption_key

Encryption key for bin files, input for compiler.

(14) simulator_img_file

Input for simulator.

The absolute path of the image you want to inferenced by simulator.

(15) emulator_img_folder

The absolute path of the image folder you want to inferenced by emulator.

(16) cmd_bin

The absolute path of command binary file, which is the input file for simulator or emulator.

(17) weight_bin

The absolute path of weight binary file, which is the input file for simulator or emulator.

(18) setup_bin

The absolute path of setup binary file, which is the input file for simulator or emulator.

(19) whether_npu_preprocess

The option for whether simulator or emulator using the same image processing as the npu uses.

If false, the parameters (20) - (25) will not be utilized.

Parameters (20) - (23) is for npu image preprocessing.

(21) raw_img_fmt

The input image format for simulator and emulator

Options: IMG, RGB565, NIR888

IMG: jpg/png/jpeg/bmp image files

RGB565: binary file with rgb565 format;

NIR888: binary file with nir888 format.

(21) radix

The radix information for the npu image process.

The formula for radix is 7 – ceil(log2 (abs_max))

For example, if the image processing method we utilize is “kneron”, which is introduced in the parameter (5). So the related image processing formula is “kneron”: RGB/256 - 0.5, and the processed value range will be (0.5, 0.5), and then

abs_max = max(abs(-0.5), abs(0.5)) = 0.5

Radix = 7 – ceil(log2(abs_max)) = 7 - (-1) = 8

(22) pad_mode

This is the option for the mode of adding paddings, and it will be utilized only when (7) keep_aspect_ratio is True.

And it has two options: 0 and 1.

0 – If the original width is too small, the padding will be added at both right and left sides equally; if the original height is too small, the padding will be added at both up and down sides equally.

1 – If the original width is too small, the padding will be added at the right side onlyl if the original height is too small, the padding will be only added at the down side.

(23) rotate

It has three options:

0 – no rotating operation

1 – rotate 90 degrees in clockwise direction

2 – rotate 90 degrees in counter-clockwise direction

(24) pCrop

The parameters for cropping image.

And it has four sub parameters.

- bCropFirstly, whether cropping the image firstly, if false, the following parameters won't be utilized, and there won't be any cropping operations.

-crop_x, cropy, the left-up cropping point coordinate.

-crop_w, the width of the cropped image.

-crop_h, the height of the cropped image.

(25) imgSize:

-width: input image width

-height: input image height

4.8.2 Fails when implement models with SSD structure.

Currently, our NPU does not support SSD like network since it has Reshape and Concat operations in the end of the model, but we do offer a work around solution to this situation.

The reason we do not support Reshape and Concat operation is that

We do offer Reshape operation capability, However, we only support regular shape transportation. Which means you could flatten your data or extract some channels form the feature map. However, NPU does not expect complex transportation. For example, in Figure 13, you could notice there is a 1x12x4x5 feature map reshapes to 1x40x6.

For Concat operation, the NPU also supports channel-based feature map concatenation. However, it does not support Concate operation based on another axis. For example, in Figure 14, The concatenation on based on axis 1 and the following concatenation is based on axis 2.

The workaround we will offer is that deleting these Reshape and Concat operations and enable make the model to a multiple outputs model since they are in the end of the models and they do not change the output feature map data. So converted model should look like this as in Figure 15.

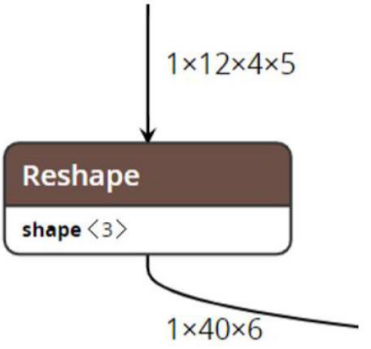


Figure 13 Invalid Reshape operation

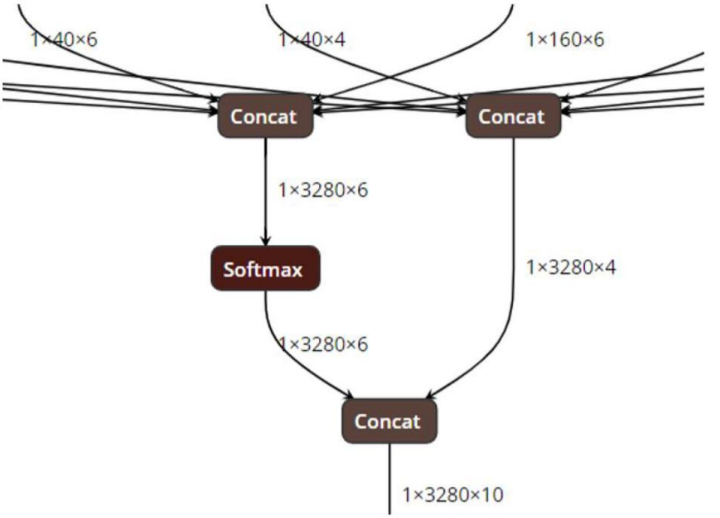


Figure 14 Invalid Concat operation

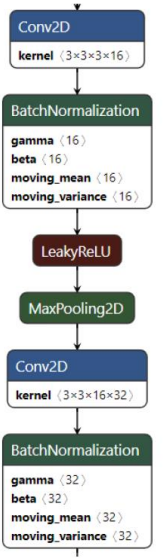


Figure 15 Valid SSD Model

4.8.3 Fails in the step of FpAnalyser

When it shows the log "mystery", it means there are some customized layers in the model you input, which are not support now;

When it shows the log "start datapath analysis", you need to check whether you input the proper image preprocess parameters.

4.8.4 Other unsupported models

This version of SDK doesn't support the models in the following situations:

(1) Have customized layers.

4.8.5 The functions KDP520 NPU supports

Layers/Modules	Functions/Parameters	Spec.
Convolution	Convolution kernel dimentison:	1x1 up to 11x11
	Stride	1,2,4
	Padding:	0-15
	Depthwise Conv	Yes
	Deconvolution	Use Upsampling + Conv
Pooling	Max pooling 3x3	stride 1,2,3
	Max pooling 2x2	stride 1,2
	Ave Pooling 3x3	stride 1,2,3
	Ave Pooling 2x2	stride 1,2
	global ave pooling	Support
	global max pooling	Support
Activation	ReLu	Support
	Leaky ReLU	Support
	PReLU	Support
	ReLU6	Support
Other processing	Batch Normalization	Support
	Add	Support
	Concatenation	Support
	Dense/Fully Connected	Support
	Flatten	Support

4.8.6 What's the meaning of simulator's output?

estimate FPS float => average Frame Per Second

total time => total time duration for single image inference on NPU

MAC idle time => time duration when NPU MAC engine is waiting for weight loading or data loading

MAC running time => time duration when NPU MAC engine is running

average DRAM bandwidth => average DRAM bandwidth used by NPU to complete inference

total theoretical convolution time => theoretically minimum total run time of the model when MAC efficiency is 100%

MAC efficiency to total time => time ratio of the theoretical convolution time to the total time

4.8.7 How to configure the batch_compile_input_params.json?

By following the above instructions, the batch_compile_input_params.json will be saved in Interactive Folder.

Please do not change the parameters' names.

The parameters in batch_compile_input_params.json are:

(1) input_image_folder

The absolute path of input image folder for fpAnalyser. Since that batch-compile can compile more than one models together, the order of the input_imgae_folder is related to the order of input_onnx_file.

(2) img_channel

Options: L, RGB

The channel information after the input image is preprocessed. L means single channel. Input for fpAnalyer. Same as (1), the order of the img_channel is related to the order of input_onnx_file.

(3) model_input_width

The width of the model input size.Input for fpAnalyer. Same as (1), the order of the model_input_width is related to the order of input_onnx_file.

(4) model_input_height

The height of the model input size. Same as (1), the order of the model_input_height is related to the order of input_onnx_file.

(5) img_preprocess_method

Options: kneron, tensorflow, yolo, caffe, pytorch. Same as (1), the order of the img_preprocess_method is related to the order of input_onnx_file.

The image preprocess methods, input for fpAnlayer, and the related formats are following:

"kneron": RGB/256 - 0.5,

"tensorflow": RGB/127.5 - 1.0,

"yolo": RGB/255.0

"pytorch": (RGB/255. -[0.485, 0.456, 0.406]) / [0.229, 0.224, 0.225]

"caffe"(BGR format) BGR - [103.939, 116.779, 123.68]

(6) input_onnx_file

The absolute path of the onnx file, which works as the input file for fpAnalyser. Since that the batch-compile can compile more than one models at a time. The order of the model in the array of input_onnx_file decides the model binary's order in all_models.bin

(7) keep_aspect_ratio

Options: True, False

Indicates whether or not to keep the aspect ratio.

(8) command_addr

Address for command, input for compiler.

(9) weight_addr

Address for weight, input for compiler.

(10) sram_addr

Address for sram, input for compiler.

(11) dram_addr

Address for dram, input for compiler.

(12) whether_encryption

Option: Yes, No

Whether add encryption on the bin files generated by compiler, input for compiler.

(13) encryption_key

Encryption key for bin files, input for compiler.

(14) model_id_list

The list of model id information

(15) model_version_list

The list of model version information.

4.8.8 What's the meaning of the output files of batch-compile?

The result of 3.7 FpAnalyser and Batch-Compile is generated at a folder called batch_compile at Interactive Folder, and it has two sub-folders called fpAnalyser and compiler.

In fpAnalyser subfolder, it has the folders with name format as input_img_txt_X, which contains the .txt files after image preprocessing. The index X is the related to the order of model file in FAQ question 7 (6), which means the folder input_img_txt_X is number X model's preprocess image text files.

In compile subfolder, it will have the following files: all_model.bin, fw_info.bin, temp_X_ioinfo.csv. The X is still the order of the models.

- all_model.bin and fw_info.bin is for firmware to use;

-temp_X_ioinfo.csv contains the information that cpu node and output node.

If you find the cpu node in temp_X_ioinfo.csv, whose format is "c,**,**", you need to implement and register this function in SDK.

4.8.9 How to use customized methods for image preprocess?

(1) Configure the input_params.json, and fill the value of "img_preprocess_method" as "customized";

(2) Edit the file /workspace/scripts/img_preprocess.py, search for the text "#this is the customized part" and add your customized image preprocess method there.

Chapter 5 Firmware Management

5.1 Update Firmware

Use the following steps and commands to update the firmware.

Step 1: Install packages

```
$ sudo apt-get install cmake

$ sudo apt-get install libusb-1.0-0-dev

$ sudo apt-get install g++
```

Step 2

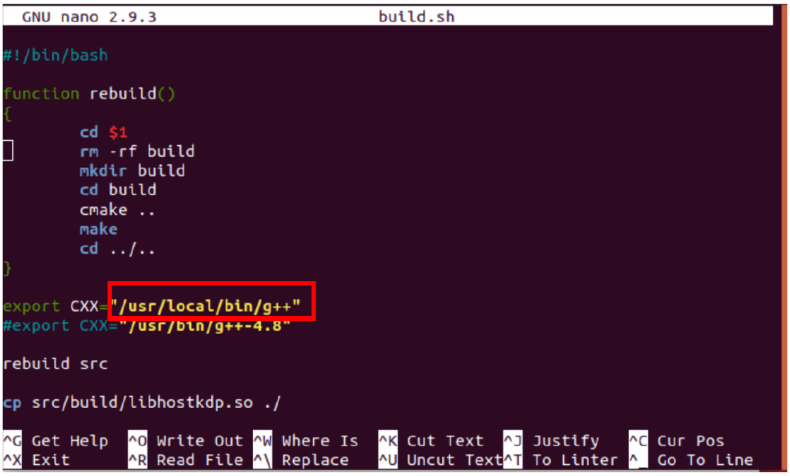
```
$ cd kl520_sdk_<version>/host_lib
```

Step 3: Change g++ file path in build.sh

```
$ which g++
```

For example: /usr/bin/g++

```
$ nano build.sh
```




```
$ sudo chmod +x build.sh
```

```
$ sudo ./build.sh

$ cd kl520.sdk.<version>/host_lib/example/build
```

Program udt_fw

Program udt_fw can be used to test the update firmware feature. The firmware files are in "../test_image/ota/work1". An alternate directory"work2"exists in the same directory, which could be used for testing of the switch of two banks.

Usage: udt_fw fw_id (0 – no operation, 1 – scpu, 2 - ncpu)

For example:

To run update scpu firmware

```
$ sudo ./udt_fw 1
```

To run update ncpu firmware

```
$ sudo ./udt_fw 2
```

Note: After firmware update finishes, the KL520 is doing reset. The KL520 needs to be restarted manually.

5.2 General Model Firmware

Step 1: Go to terminal

Ener the following command:

```
cd kl520_sdk_<version>/ota
```

Step 2: Copy fw_info.bin & all_models.bin in 3.6.2 BatchCompile to kl520_sdk_<version>/ota

Step 3: Run below command line(fw_info.bin & all_models.bin generated from Batch Compile

[refer Page 5])

```
sudo chmod +x gen_ota_binary_for_linux

./gen_ota_binary_for_linux -model fw_info.bin all_models.bin
model_ota.bin
```

The model file model_ota.bin is generated in kl520_sdk_<version>/ota

5.3 Model Update

Program udt_md

Program udt_md can be used to test the update model feature. The model file is

"kl520_sdk_<version>/host_lib/example/test_image/ota model_ota.bin ".

Step 1:

Replace model_ota.bin in test_image file by model_ota.bin that generated from 2. Generate model firmware

Usage: udt_md model_id (0 – no operation, other value – model id)

Step 2:

To run update model with model id 1

```
$ sudo ./udt_md 1
```

To run update model with empty operation

```
$ sudo ./udt_md 0
```

Note：After update model finishes, the KL520 is doing reset. The KL520 needs to be re-started manually, either by SPI or by JTAG. After the system is started successfully, KL520 can send back the response.

APIs used:

kdp_update_model()