Khadas Software Neural Network API Documentation

Submodules

ksnn.api module

class **ksnn.api.KSNN**(*board=None*)

Bases: `object`

Neural Network control interfance

Class KSNN is the control interface for Neural Network,

all NPU-related functions and operations are included in this class.

board

Board model. Board model list: VIM3/VIM3L

get_nn_version()

Print Neural Network Api version

Parameters: **None** –

Returns: version

Return type: string

nn_get_output_tensor_info(*num*)

Get output tensor info

Parameters: **num** – Which output layer

Returns: npu_tensor

Return type: class

nn_get_outputs(*num=1*, *out_format=out_format.OUT_FORMAT_FLOAT32*)

Get outputs data after run Neural Network

Parameters:

- **num** – Number of output layers. default is 1.

- **out_format** – Data format of output data (OUT_FORMAT_UINT8/OUT_FORMAT_INT8/ OUT_FORMAT_INT16/OUT_FORMAT_FLOAT32(default)).

Returns: List of numpy arrays

Return type: list()

nn_inference(*cv_img*, *platform=None*, *reorder='0 1 2'*, *num=1*, *out_format=out_format.OUT_FORMAT_FLOAT32*)

nn_inference implements a unified interface from input to output

Parameters:

- **cv_img** – Mat format data
- **platform** – Your origin model platform (TENSORFLOW/CAFFE/PYTROCH/DARKNET/ONNX/KERAS)
- **reorder** – Channel order('0 1 2'/'2 1 0')
- **num** – Number of output layers. default is 1.
- **out_format** – Data format of output data (OUT_FORMAT_UINT8/OUT_FORMAT_INT8/ OUT_FORMAT_INT16/OUT_FORMAT_FLOAT32(default)).

Returns: List of numpy arrays

Return type: list()

nn_init(*c_lib_p=None*, *nb_p=None*)

Create Neural Network

Parameters:

- **c_lib_p** – (Only valid for VIM3/VIM3L) The path for your C static librarys
- **nb_p** – (Only valid for VIM3/VIM3L) The path for your nb file

Returns: ksnn_stat

Return type: class

nn_run()

Run neural network

Parameters: **None** –

Returns: ksnn_stat

Return type: class

nn_set_inputs(*img*, *platform=None*, *reorder='0 1 2'*)

Convert the data and set it into neural network

Parameters:

- **img** – Mat format data
- **platform** – Your origin model platform (TENSORFLOW/CAFFE/PYTROCH/DARKNET/ONNX/KERAS)
- **reorder** – Channel order(‘0 1 2’/’2 1 0’)

Returns: ksnn_stat

Return type: class

ksnn.api.file_exist_judgment(*file_path*)

Verify that the file exists

ksnn.types module

class ksnn.types.ksnn_board(*value*)

Bases: `enum.Enum`

Support Board List

BOARD_UNKNOWN= *0*

BOARD_VIM3= *1*

BOARD_VIM3L= *2*

class ksnn.types.ksnn_stat(*value*)

Bases: `enum.Enum`

Neural Network stat Enum class

STAT_FAIL= *1*

STAT_SUCCESS= *0*

class ksnn.types.out_format(*value*)

Bases: `enum.Enum`

Support output format

OUT_FORMAT_FLOAT32= *3*

> OUT_FORMAT_INT16= *2*

> OUT_FORMAT_INT8= *1*

> OUT_FORMAT_UINT8= *0*

Module contents