

---

# ksnn package

## Subpackages

## Submodules

## ksnn.api module

---

*class* `ksnn.api.KSNM(board=None)`

Bases: `object`

Neural Network control interfance

Class KSNM 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(tensor=1, output\_format=output\_format.OUT\_FORMAT\_FLOAT32)**

Get outputs data after run Neural Network

**Parameters:**    • **tensor** – Number of output layers. default is 1.

- **output\_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', input\_tensor=1, output\_tensor=1, output\_format=output\_format.OUT\_FORMAT\_FLOAT32*)

nn\_inference implements a unified interface from input to output

- Parameters:**
- **cv\_img** – Mat format data list
  - **platform** – Your origin model platform  
(TensorFlow/Caffe/PyTorch/DarkNet/ONNX/Keras)
  - **reorder** – Channel order('0 1 2'/'2 1 0')
  - **input\_tensor** – Number of input layers. default is 1.
  - **output\_tensor** – Number of output layers. default is 1.
  - **output\_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**(*library=None, model=None, level=0*)

Create Neural Network

- Parameters:**
- **library** – (Only valid for VIM3/VIM3L) The path for your C static libraries
  - **model** – The path for your model file.(VIM3 is nbg file)
  - **level** – Information printing level (0(default)/1/2)

**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', tensor=1)
```

Convert the data and set it into neural network

**Parameters:**

- **img** – Mat format data
- **platform** – Your origin model platform (TensorFlow/Caffe/PyTorch/DarkNet/ONNX/Keras/TFLite)
- **reorder** – Channel order('0 1 2'/'2 1 0')
- **tensor** – the input tensor numbers

**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.output_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**