

node-red-contrib-edgetpu-inference 1.0.1

This package contains Node-RED nodes which taking advantage of Shenzhou TPU to infer using AI models. There are also nodes for GUI presentation of result and system performance in this package.

Install

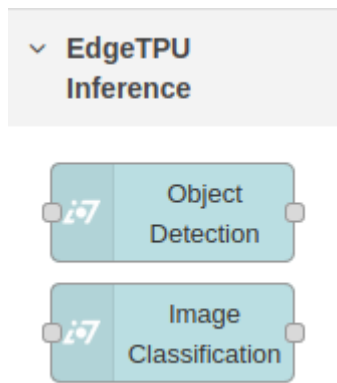
Run the following npm command in your Node-RED user directory (typically ~/.node-red):

```
npm install --prefix=~/.node-red node-red-contrib-edgetpu-inference
```

Category

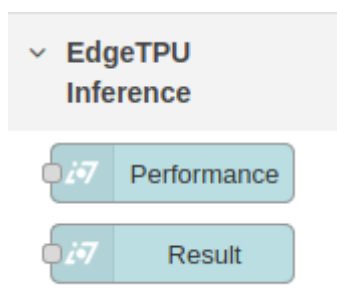
- Inference Node

Find the inference node in category " EdgeTPU Inference " as follow:



- Result Node and Performance Node

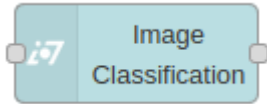
Find the inference node in category " EdgeTPU Inference " as follow:



Usage

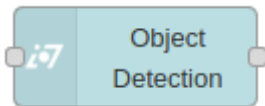
- Inference Node

> "Image Classification" node:



Run image classification models with specified TPUs

> "Object Detection" node:



Run object detection models with specified TPUs

- Node Properties:

Below is the explanation of the properties:

1. **Input Type:** The source input format, including the paths of image file, URL link, video file and camera device
2. **TPU Type:** Support M.2 edge TPU type at current
3. **TPU ID:** Specify the index of edge TPU
4. **Model Path:** The path for model file
5. **Label Path:** The path for label file of model
6. **Image Classification API parameter:**
 - o threshold : The score threshold for results. All returned results have a score greater-than-or-equal-to this value
 - o top_k : The maximum number of results

Edit Image Classification node

Delete

Cancel

Done

⚙ Properties

⚙

📄

🖼

🔑 Name

Name of the node

🔑 Input Type

Image

🔑 TPU Type

ShenZhou (PCIe)

🔑 TPU ID

2

🔑 Model Path

./node_modules/node-red-contrib-edgetpu-inferen

🔑 Label Path

./node_modules/node-red-contrib-edgetpu-inferen

☑ Image Classification API Parameters

threshold

0.1

top_k

3

🔄 Control

✓ Output Image

✓ Video Loop

✓ Parse Parti cular Label

Label Name

person

✓ Speed Up

Downscale(%)

10

Before/After Inf.

Before

7. Object Detection API parameter:

o threshold : The score threshold for results. All returned results have a score greater-than-or-equal-to this value.

Edit Object Detection node

Delete
Cancel
Done

Properties

Name
EdgeTPU 0

Input Type
Video

TPU Type
ShenZhou (PCIe)

TPU ID
0

Model Path
./node_modules/node-red-contrib-edgetpu-inference/test_dat

Label Path
./node_modules/node-red-contrib-edgetpu-inference/test_dat

Object Detection API Parameters

threshold
0.5

Control

Output Image

Video Loop

Parse Particular Label

Label Name
Person

Speed Up

Downscale(%)
50

Before/After Inf.
Before

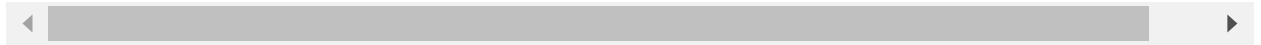
Note: The API parameter can reference the following link

- o <https://coral.ai/docs/edgetpu/api-intro/#edge-tpu-api-overview>
(<https://coral.ai/docs/edgetpu/api-intro/#edge-tpu-api-overview>)

8. Control

- o Output Image : Output base64 image
- o Video Loop : Play video looply
- o Parse Particular Label: Send image contents when get the particular label. If don't get the particular label, the node don't send the image contents.
- o Speed Up: Reduce the base64 covert time by adjust the input frame resolution

- Downscale(%): scale down the resolution
- Before/After inf: scale down the input frame before/after inference by edget



- Input and Output data Formats:

1. Input data format to inference node:

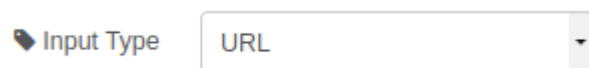
When the inference using a edgetpu model is performed, you need to pass the corresponding msg.payload to the inference node. The msg.payload would be a string of path about image or frame sources.

Source Type	Payload format	Example
Image	Strings	"/home/asus/Desktop/test.jpg"
URL Streaming server	Strings	"http://127.0.0.1:8080/?action=stream (http://127.0.0.1:8080/?action=stream)"
Video	Strings	"/home/asus/Desktop/test.mp4"
Local Camera	Strings	"0"
Stop Inference node	Strings	"STOP" or "stop"
Pause Inference node	Strings	"PAUSE" or "pause"

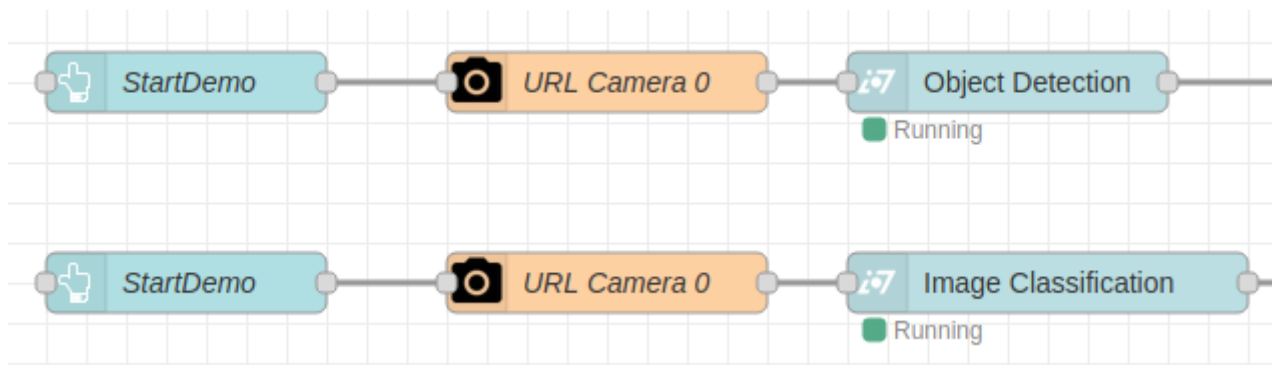
An example of input to inference node for url streaming server:

```
{
  payload: "http://127.0.0.1:8080/?action=stream"
}
```

You also need to select the "input type" item to URL on inference node:



1.1 Input data format to inference node:

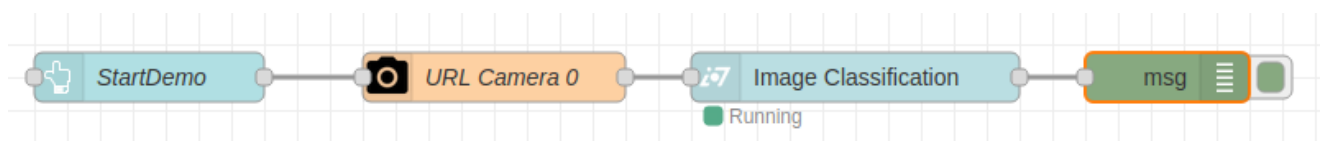


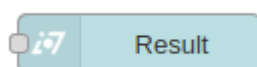
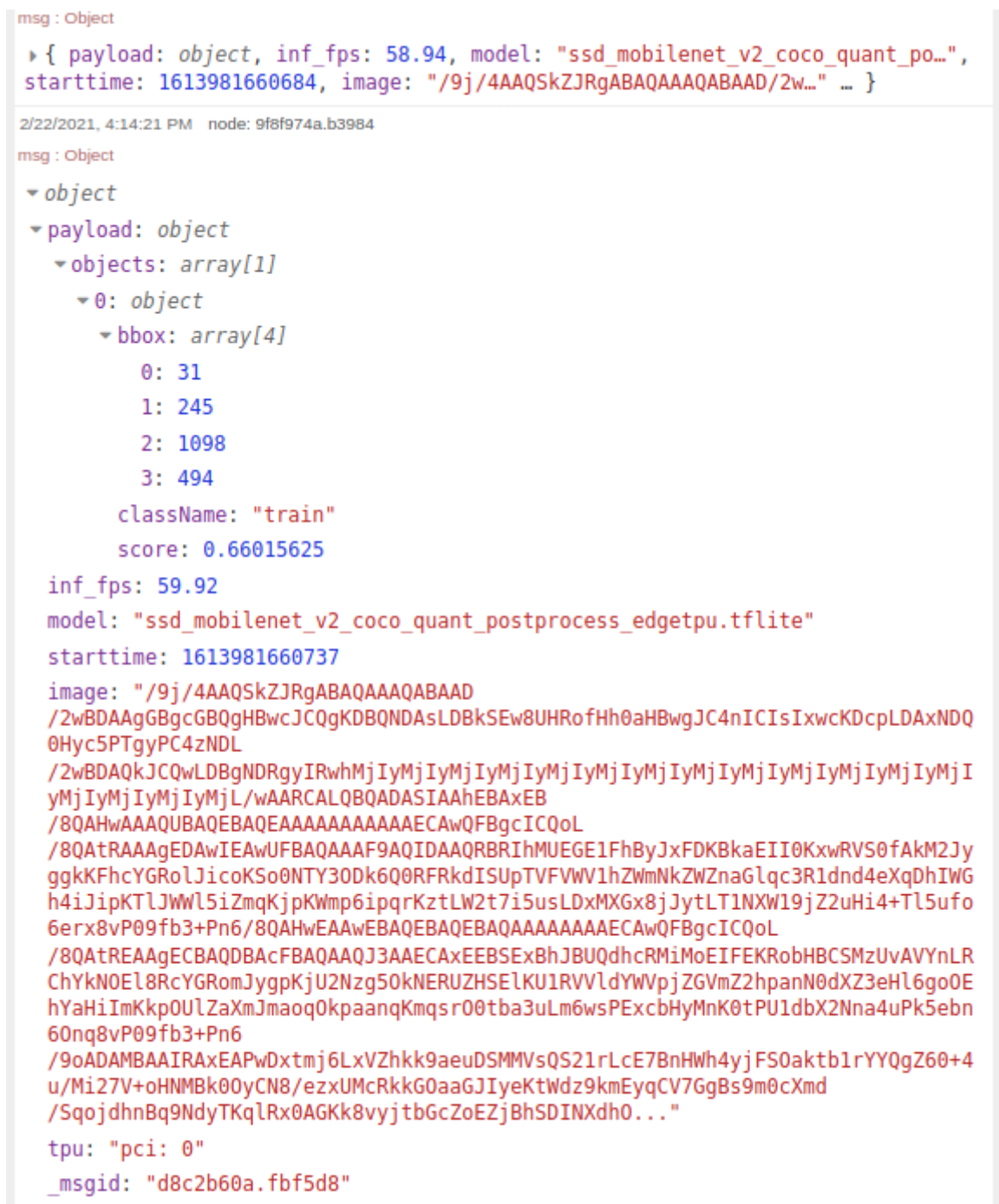
2. Output data format from inference node::

2.1 SZ Image Classification node output json format:

Output	Format	Description
className	Strings	Class Name category
score	Integer	The confidence of the inference result
inf_fps	Integer	The FPS about TPU inference for a frame
starttime	Integer	Inference node start time(Millionseconds)
image	Strings	Base64 format strings (Output Image item is selected and then the image would be transferred)
model	Strings	The filename of model that inference node used
tpu	Strings	The TPU used by inference node

Reference the Results on Node-red debug message:





1. Group: Select which group on dashboard and show the widgets

2. Size : sets the basic geometry of the grid layout in pixels
3. Label : Show the topic on the dashboard
4. ip: The localhost or ip address that is provided the device to connect.
5. Name : Config and show the name on the node

Edit Result node

Delete Cancel Done

Properties

Group [ASUS IoT Object Detection] EdgeTPU I

Size auto

Label

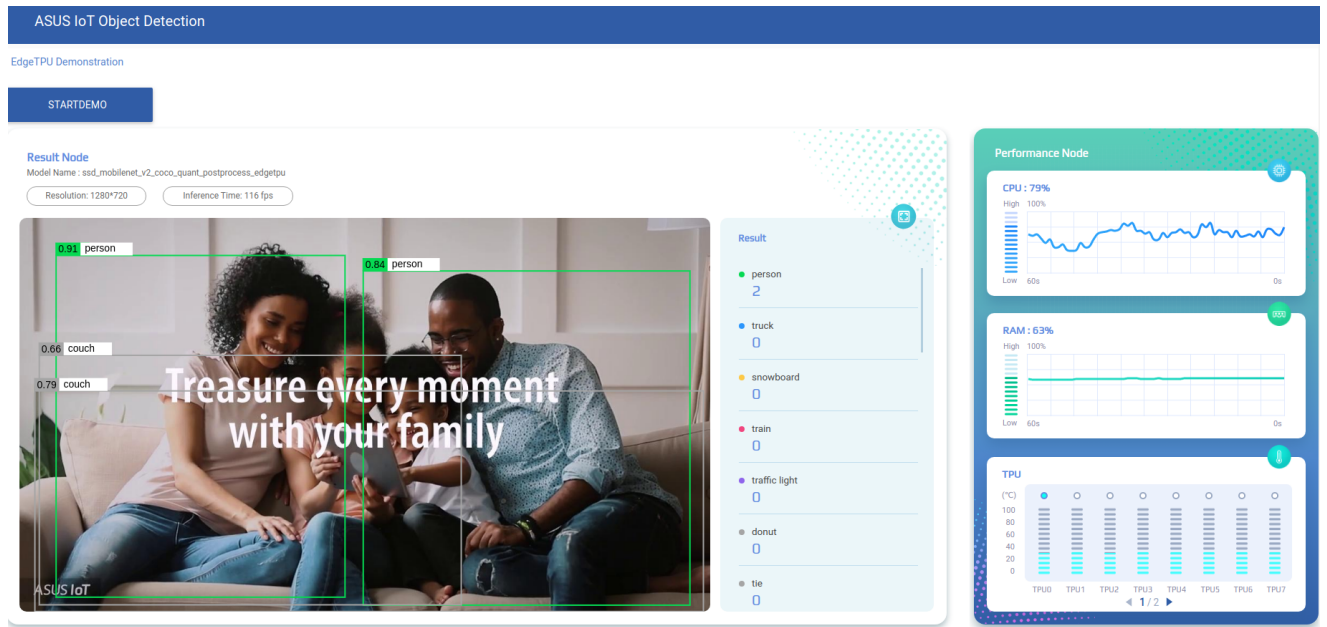
Ip 127.0.0.1

Name

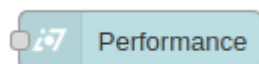
- Output

Reference the following image about the node outputs:

1. Default mapping to Group
2. Label mapping to the Label config of properties
3. The colors of bounding-box are classified with the object recognition :
 - The colors is mapped by the order of the object detection
 - Green is the first recognized object
 - Gray is the sixth recognized object or after the sixth object
4. The right filed (Total): show the quantity of classification object



• Performance node



Show the following system information:

1. The temperature of edge TPUs
2. CPU usage
3. RAM usage

- Node Properties:

1. Group: Select which group on dashboard and show the widgets
2. Size: Sets the basic geometry of the grid layout in pixels
3. ip: The localhost or ip address that is provided the remote device to connect to.
4. Name: Config and show the name on the node

Edit Performance node

Delete

Cancel

Done

⚙️ Properties

⚙️

📄

🖨️

📊 Group

[ASUS IoT Object Detection] EdgeTPU I ▾

✎

🖼️ Size

auto

🌐 Ip

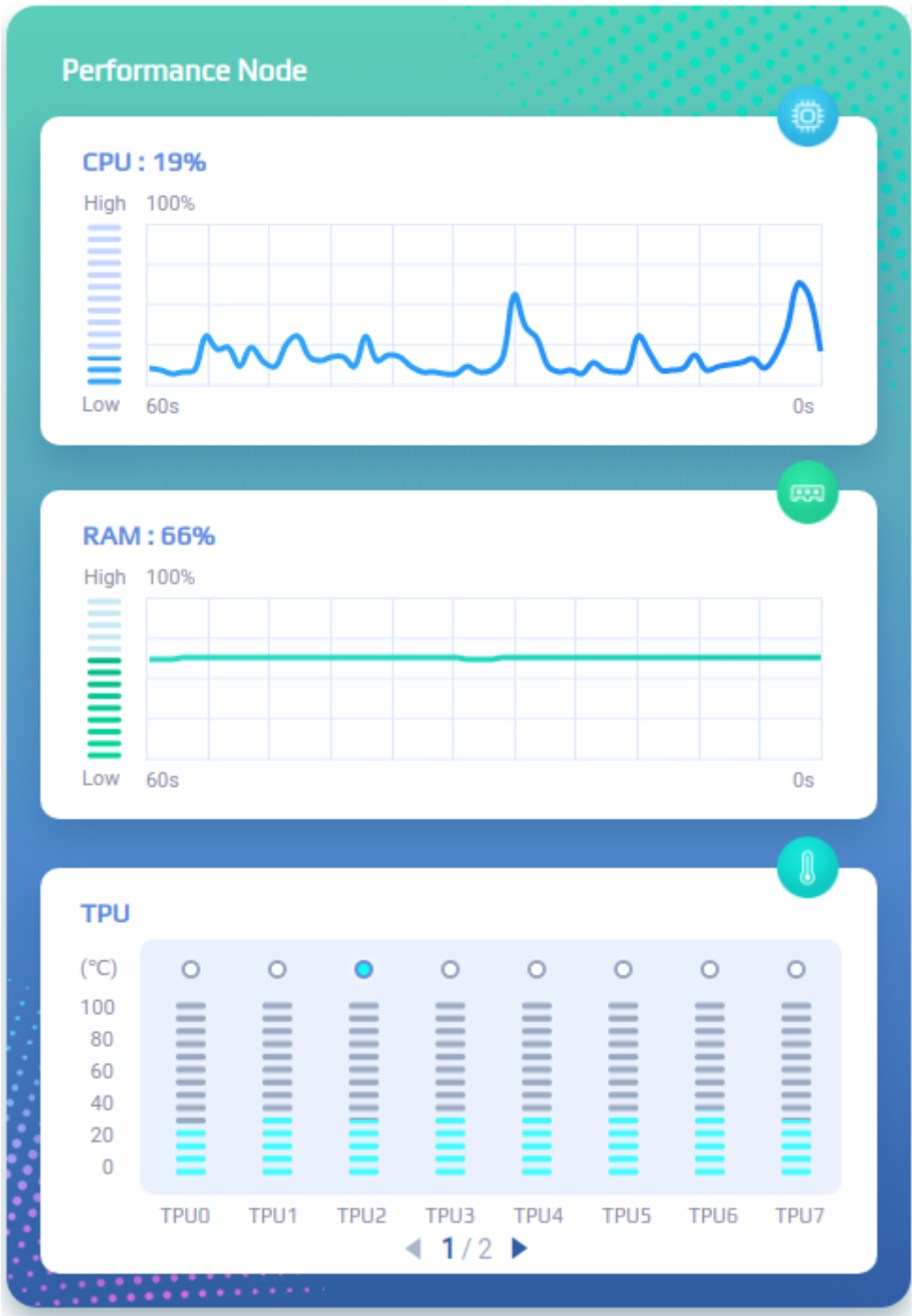
127.0.0.1

🏷️ Name

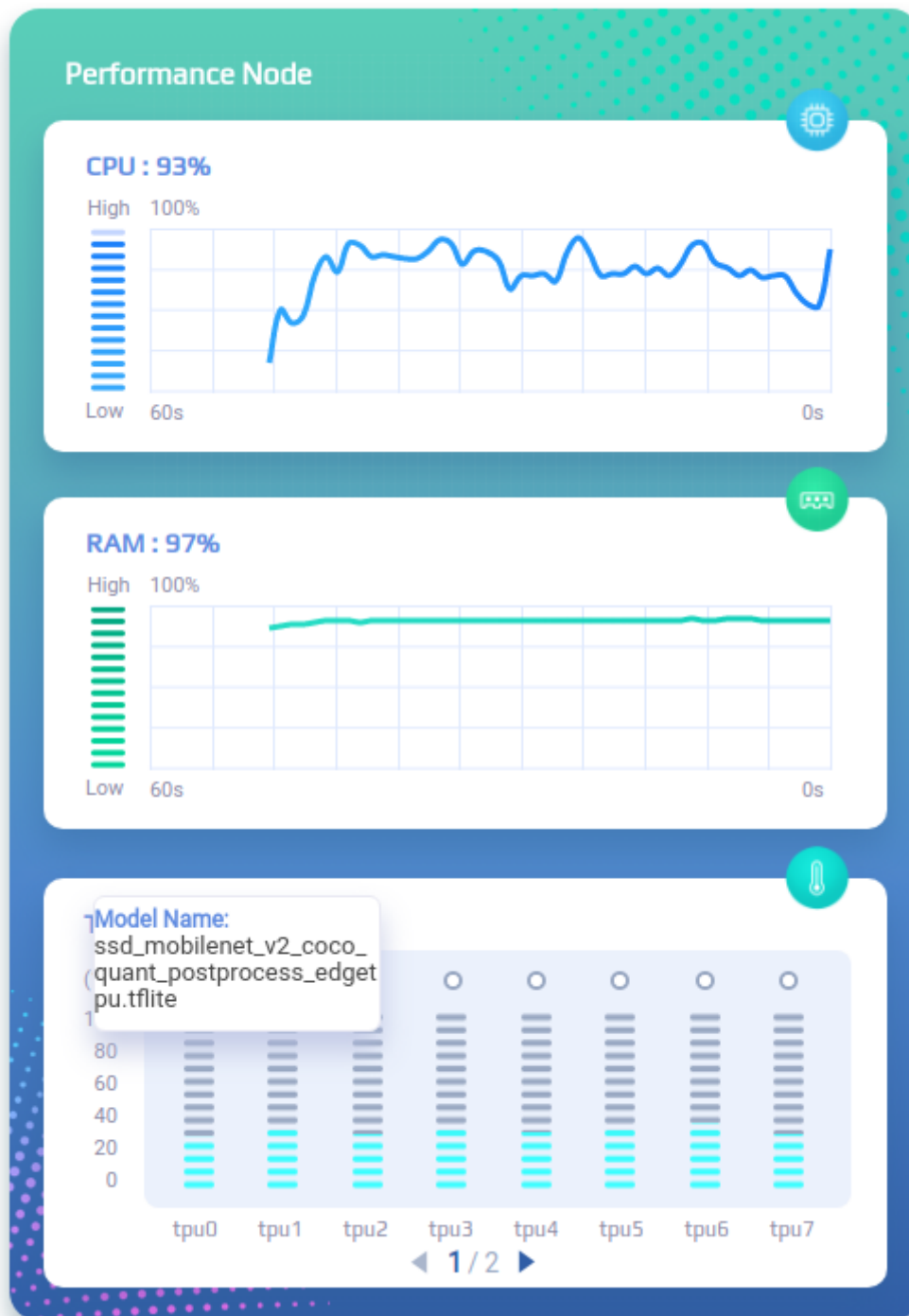
- Output

1. Default1 mapping to Group

2. The following informaion : CPU usage, RAM usage and TPU temperature

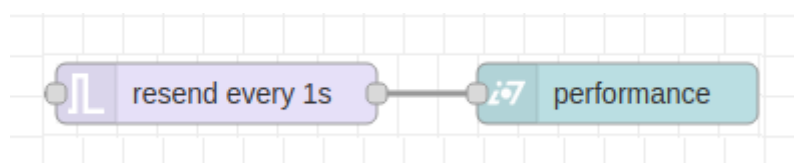


3. When the mouse cursor rests on the tpu bar, the model name in use will be shown



- Example flow

Get an "inject node" and then connect to performance node:



Set the "Repeat" field of "inject node" to "interval" every 1 second and then the performance node will update the result per second

+ add

☐ Inject once after seconds, then

Repeat

interval

▼

every

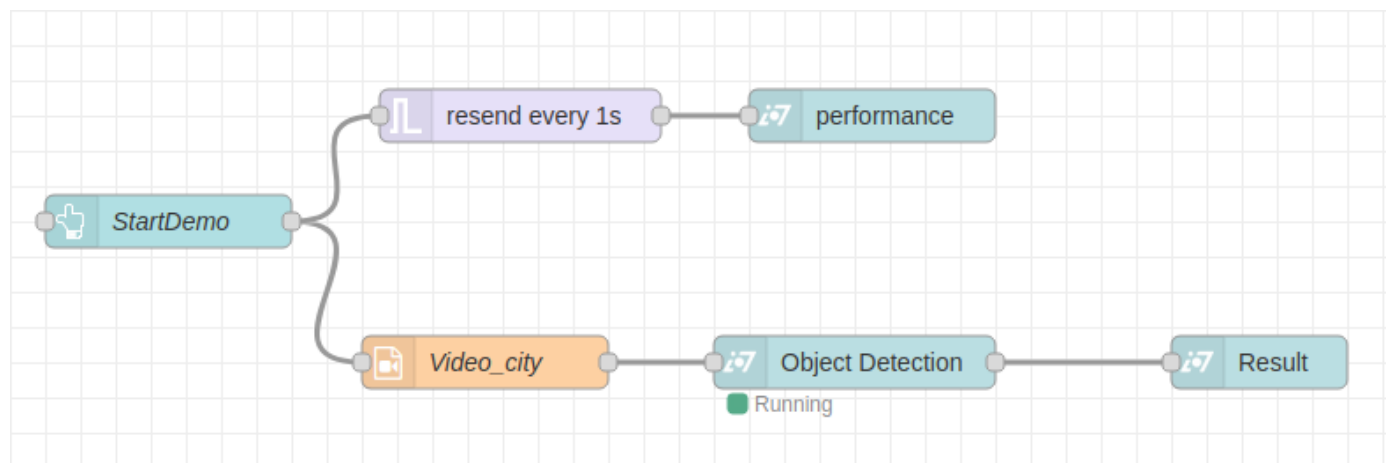
1

▲▼

seconds

▼

node-red-contrib-edgetpu-inference Example Flow



ASUS IoT Object Detection

EdgeTPU Demonstration

STARTDEMO

Result Node

Model Name : ssd_mobilenet_v2_coco_quant_postprocess_edgetpu

Resolution: 1280*720

Inference Time: 116 fps

Result

- person 2
- truck 0
- snowboard 0
- train 0
- traffic light 0
- donut 0
- tie 0

Performance Node

CPU: 79%

RAM: 63%

TPU

```
[
  {
    "id": "57023ee8.030fa",
    "type": "tab",
    "label": "ASUS IoT Object Detection",
    "disabled": false,
    "info": ""
  },
  {
    "id": "e9ce2c96.042b8",
    "type": "ui_group",
    "z": "",
    "name": "EdgeTPU Demonstration",
    "tab": "2b62772c.311188",
    "order": 3,
    "disp": true,
    "width": "35",
    "collapse": false
  },
  {
    "id": "2b62772c.311188",
    "type": "ui_tab",
    "z": "",
    "name": "ASUS IoT Object Detection",
    "icon": "dashboard"
  },
  {
    "id": "7f03a96d.192848",
    "type": "ui_base",
    "theme": {
      "name": "theme-light",
      "lightTheme": {
        "default": "#0094CE",
        "baseColor": "#2e5ea8",
        "baseFont": "-apple-system,BlinkMacSystemFont,Segoe UI,Roboto,Oxygen-S",
        "edited": true,
        "reset": false
      },
      "darkTheme": {
        "default": "#097479",
        "baseColor": "#097479",
        "baseFont": "-apple-system,BlinkMacSystemFont,Segoe UI,Roboto,Oxygen-S",
        "edited": false
      },
      "customTheme": {
        "name": "Untitled Theme 1",
        "default": "#4B7930",
        "baseColor": "#4B7930",
        "baseFont": "-apple-system,BlinkMacSystemFont,Segoe UI,Roboto,Oxygen-S"
      },
      "themeState": {
        "base-color": {
          "default": "#0094CE",
          "value": "#2e5ea8",

```

```

        "edited": true
    },
    "page-titlebar-backgroundColor": {
        "value": "#2e5ea8",
        "edited": false
    },
    "page-backgroundColor": {
        "value": "#fafafa",
        "edited": false
    },
    "page-sidebar-backgroundColor": {
        "value": "#333333",
        "edited": false
    },
    "group-textColor": {
        "value": "#5384d0",
        "edited": false
    },
    "group-borderColor": {
        "value": "#ffffff",
        "edited": false
    },
    "group-backgroundColor": {
        "value": "#ffffff",
        "edited": false
    },
    "widget-textColor": {
        "value": "#111111",
        "edited": false
    },
    "widget-backgroundColor": {
        "value": "#2e5ea8",
        "edited": false
    },
    "widget-borderColor": {
        "value": "#ffffff",
        "edited": false
    },
    "base-font": {
        "value": "-apple-system,BlinkMacSystemFont,Segoe UI,Roboto,Oxygen-
    },
    "angularTheme": {
        "primary": "indigo",
        "accents": "blue",
        "warn": "red",
        "background": "grey"
    }
},
"site": {
    "name": "Node-RED Dashboard",
    "hideToolbar": "false",
    "allowSwipe": "false",
    "lockMenu": "false",
    "allowTempTheme": "true",

```



```

        "dateFormat": "DD/MM/YYYY",
        "sizes": {
            "sx": 48,
            "sy": 48,
            "gx": 6,
            "gy": 6,
            "cx": 6,
            "cy": 6,
            "px": 0,
            "py": 0
        }
    },
    {
        "id": "acaf378c.638728",
        "type": "ui_spacer",
        "name": "spacer",
        "group": "e9ce2c96.042b8",
        "order": 2,
        "width": 26,
        "height": 1
    },
    {
        "id": "f9099855.51d4f8",
        "type": "function",
        "z": "57023ee8.030fa",
        "name": "Video_AIOT",
        "func": "\\nmsg.payload=\\\"./node_modules/node-red-contrib-edgetpu-inference/tes",
        "outputs": 1,
        "noerr": 0,
        "initialize": "",
        "finalize": "",
        "x": 350,
        "y": 460,
        "wires": [
            [
                "fc460f53.c4ceb"
            ]
        ],
        "icon": "font-awesome/fa-file-video-o"
    },
    {
        "id": "fc460f53.c4ceb",
        "type": "SZ Object Detection",
        "z": "57023ee8.030fa",
        "name": "",
        "intype": "2",
        "tputype": "0",
        "tpunum": "0",
        "modelpath": "./node_modules/node-red-contrib-edgetpu-inference/test_data/ssd",
        "labelpath": "./node_modules/node-red-contrib-edgetpu-inference/test_data/coco",
        "threshold": "0.5",
        "topk": "5",
        "keepratio": "0",
        "relativecoord": "0",
    }

```

```

        "resample": "0",
        "outimage": true,
        "loop": true,
        "isparsed": false,
        "parselabel": "person",
        "speedup": false,
        "dnsscale": "20",
        "dstime": "0",
        "x": 560,
        "y": 460,
        "wires": [
            [
                "ed840f2.3966df"
            ]
        ]
    },
    {
        "id": "795e500c.6a7cd",
        "type": "ui_button",
        "z": "57023ee8.030fa",
        "name": "StartDemo",
        "group": "e9ce2c96.042b8",
        "order": 1,
        "width": 4,
        "height": 1,
        "passthru": false,
        "label": "StartDemo",
        "tooltip": "",
        "color": "",
        "bgcolor": "",
        "icon": "",
        "payload": "",
        "payloadType": "str",
        "topic": "",
        "x": 150,
        "y": 380,
        "wires": [
            [
                "f9099855.51d4f8",
                "67a1b001.d76b6"
            ]
        ]
    },
    {
        "id": "67a1b001.d76b6",
        "type": "trigger",
        "z": "57023ee8.030fa",
        "name": "",
        "op1": "1",
        "op2": "0",
        "op1type": "str",
        "op2type": "str",
        "duration": "-1",
        "extend": false,
        "units": "s",

```

```
    "reset": "",
    "bytopic": "all",
    "topic": "topic",
    "outputs": 1,
    "x": 350,
    "y": 320,
    "wires": [
      [
        "93e3a74.887aa58"
      ]
    ]
  },
  {
    "id": "ed840f2.3966df",
    "type": "ui_result",
    "z": "57023ee8.030fa",
    "group": "e9ce2c96.042b8",
    "name": "",
    "ip": "127.0.0.1",
    "title": "",
    "order": 3,
    "resolution": "1",
    "width": 0,
    "height": 0,
    "x": 780,
    "y": 460,
    "wires": []
  },
  {
    "id": "93e3a74.887aa58",
    "type": "ui_performance",
    "z": "57023ee8.030fa",
    "group": "e9ce2c96.042b8",
    "name": "",
    "ip": "127.0.0.1",
    "title": "",
    "order": 3,
    "width": 0,
    "height": 0,
    "x": 580,
    "y": 320,
    "wires": []
  }
]
```

