node-red-contrib-edgetpu-inference `1.0.0`

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

Install

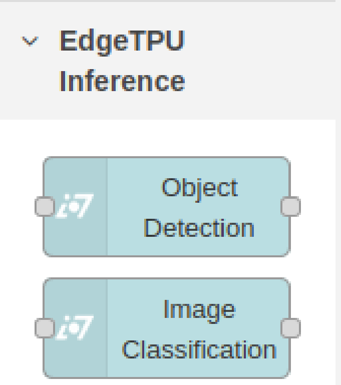
Run the following npm command in your Node-RED user directory (typically ~/.node-red):

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

Category

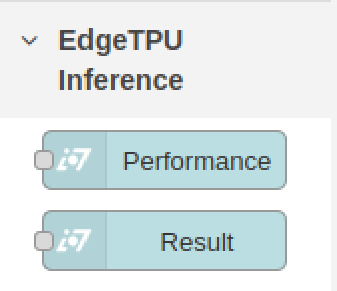
- ## Inference Node

  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:



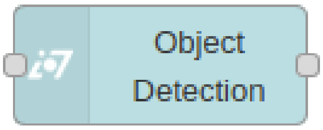
Usage

• Inference Node

> “Image Classification” node:

Run image classification models with specified TPUs

> “Object Detection” node:



Run object detection models with specified TPUs

- Node Properties:

The explanation about the items of 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

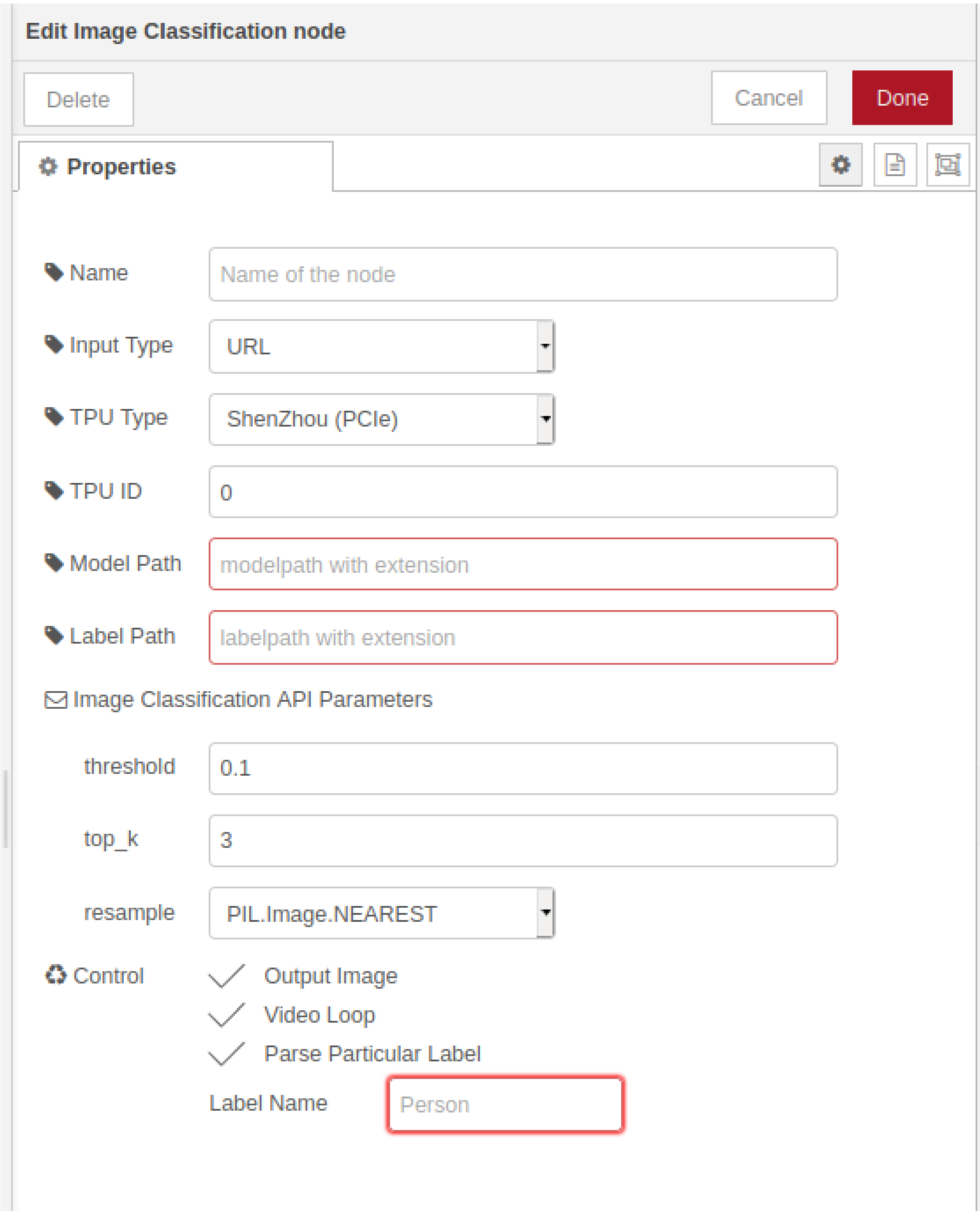
3. **TPU ID:** Select a number of edge TPU

4. **Model Path:** The edge TPU support model files path

5. **Label Path:** The label file path of model

6. **Image Classification API parameter:**
   o threshold : Minimum confidence threshold for results.
   o top_k : The maximum number of results

o resample : A resampling filter for image resizing.



Edit Image Classification node

Name: Name of the node

Input Type: URL

TPU Type: ShenZhou (PCIe)

TPU ID: 0

Model Path: modelpath with extension

Label Path: labelpath with extension

Image Classification API Parameters

threshold: 0.1

top_k: 3

resample: PIL.Image.NEAREST

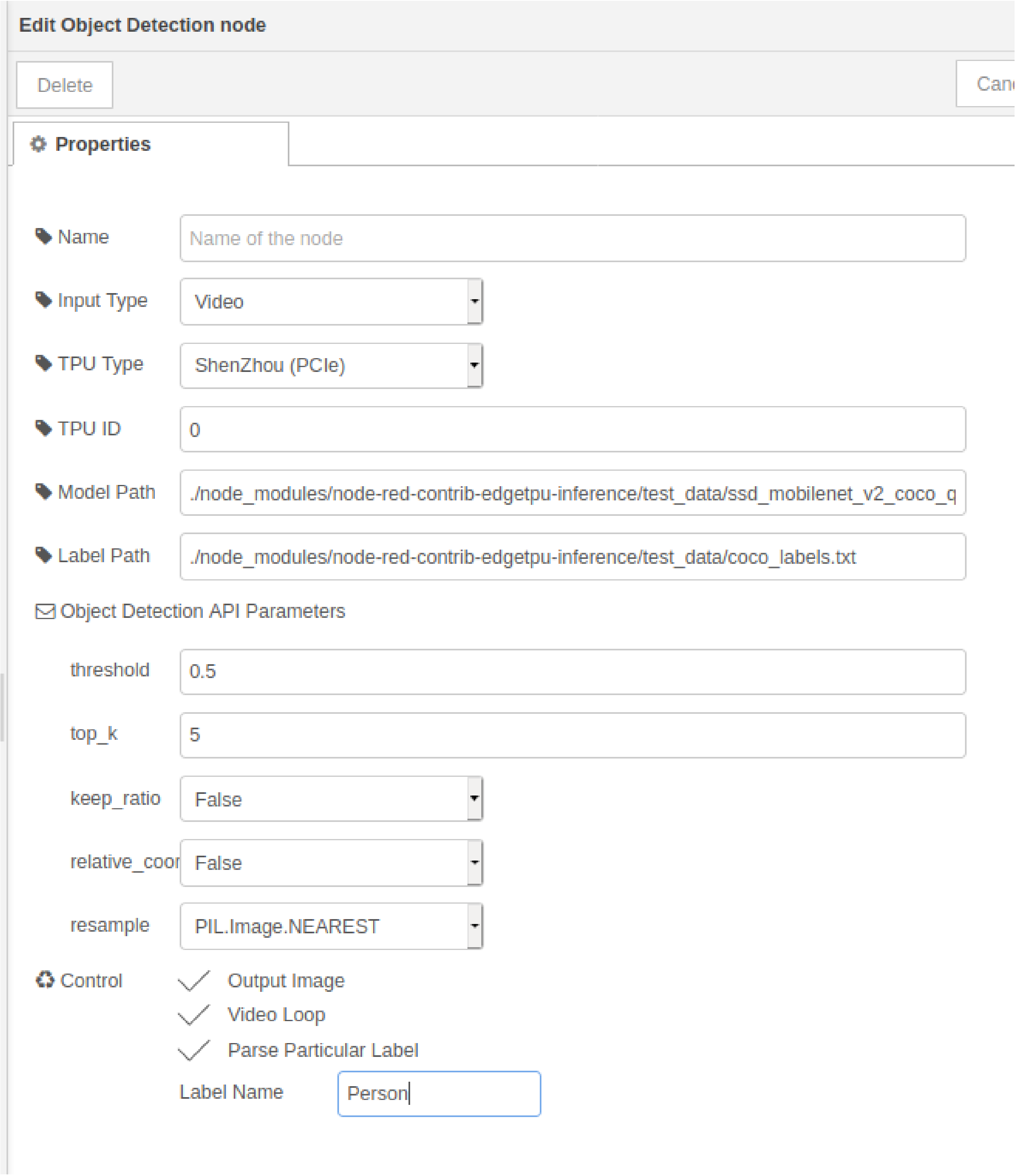
Control:
✓ Output Image
✓ Video Loop
✓ Parse Particular Label

Label Name: Person

7. **Object Detection API parameter:**
   o threshold : Minimum confidence threshold for detected objects.
   o top_k : The maximum number of detected objects to return.
   o keep_ratio : If true, keeping the image aspect ratio the same when down-sampling the image. If false, resizing and reshaping the image (without cropping) to match the input tensor's dimensions.
   o relative_coord : If true, providing coordinates as float values between 0 and 1, representing each position relative to the total image width/height. If false, providing coordinates as integers, representing pixel positions in the original image.

[0, 0] is always the top-left corner.

o resample : A resampling filter for image resizing.



Edit Object Detection node

Name: Name of the node

Input Type: Video

TPU Type: ShenZhou (PCIe)

TPU ID: 0

Model Path: ./node_modules/node-red-contrib-edgetpu-inference/test_data/ssd_mobilenet_v2_coco_q

Label Path: ./node_modules/node-red-contrib-edgetpu-inference/test_data/coco_labels.txt

Object Detection API Parameters

threshold: 0.5

top_k: 5

keep_ratio: False

relative_coor: False

resample: PIL.Image.NEAREST

Control

✓ Output Image

✓ Video Loop

✓ Parse Particular Label

Label Name: Person

Note: The API parameter can reference the following link

- 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

- **Input and Output data Formats:**

1. Input data format to inference node:

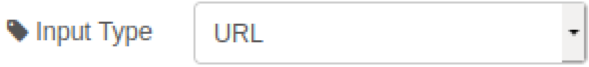
When the infernce using a edgetpu model is performed, you need to pass the corresponding msg.payload to the inference node. The msg.payload would be a string of path about image or frame sources.

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:

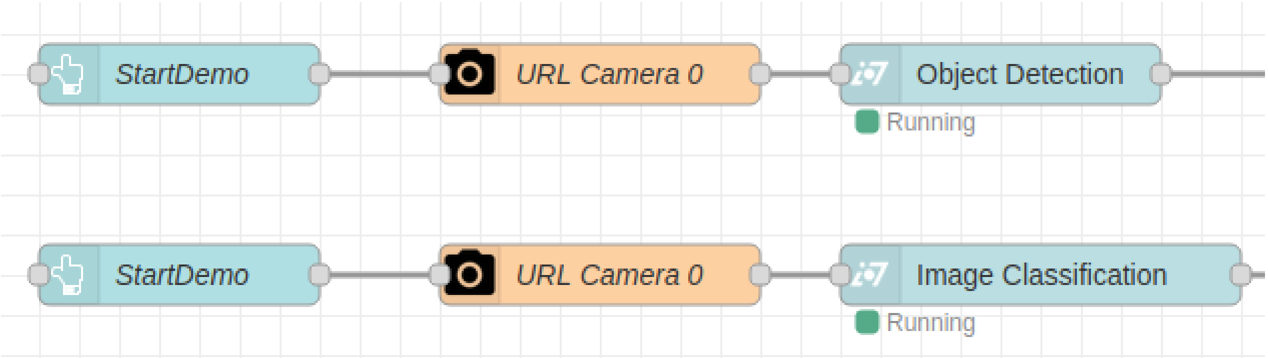
```
{
     payload: “http://127.0.0.1:8080/?action=stream”
}
```

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



Input Type: URL

1.1 Input data format to inference node:

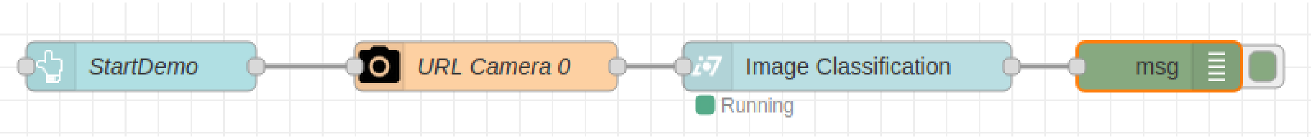


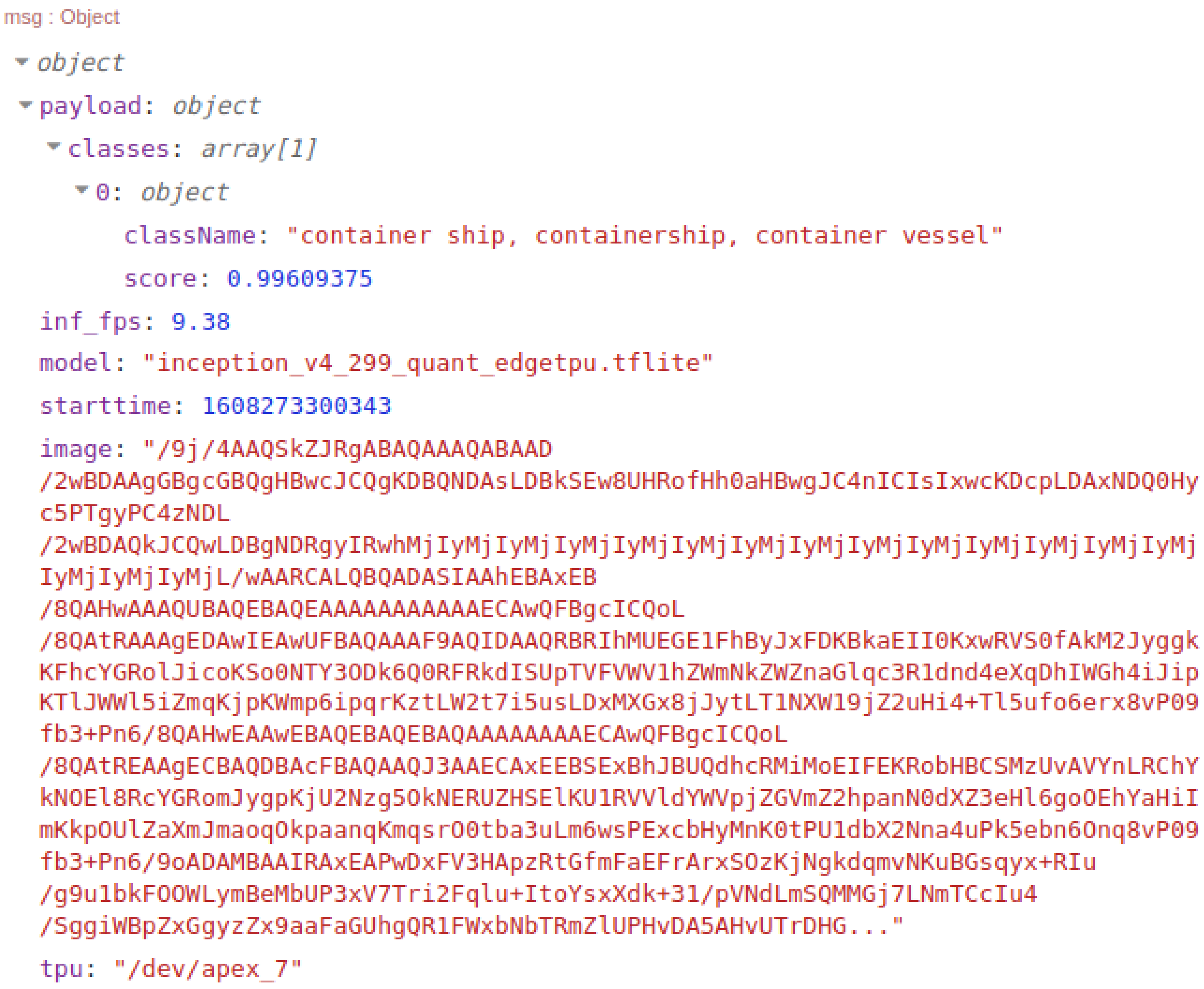
2. Output data format from inference node::

2.1 SZ Image Classification node output json format:

Output	Format	Description
className	Strings	Class Name category
score	Integer	The percent about 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 transfer)
model	Strings	Inference node import the model's filename
tpu	Strings	TPU node be used by inference node

Reference the Results on Node-red debug message:



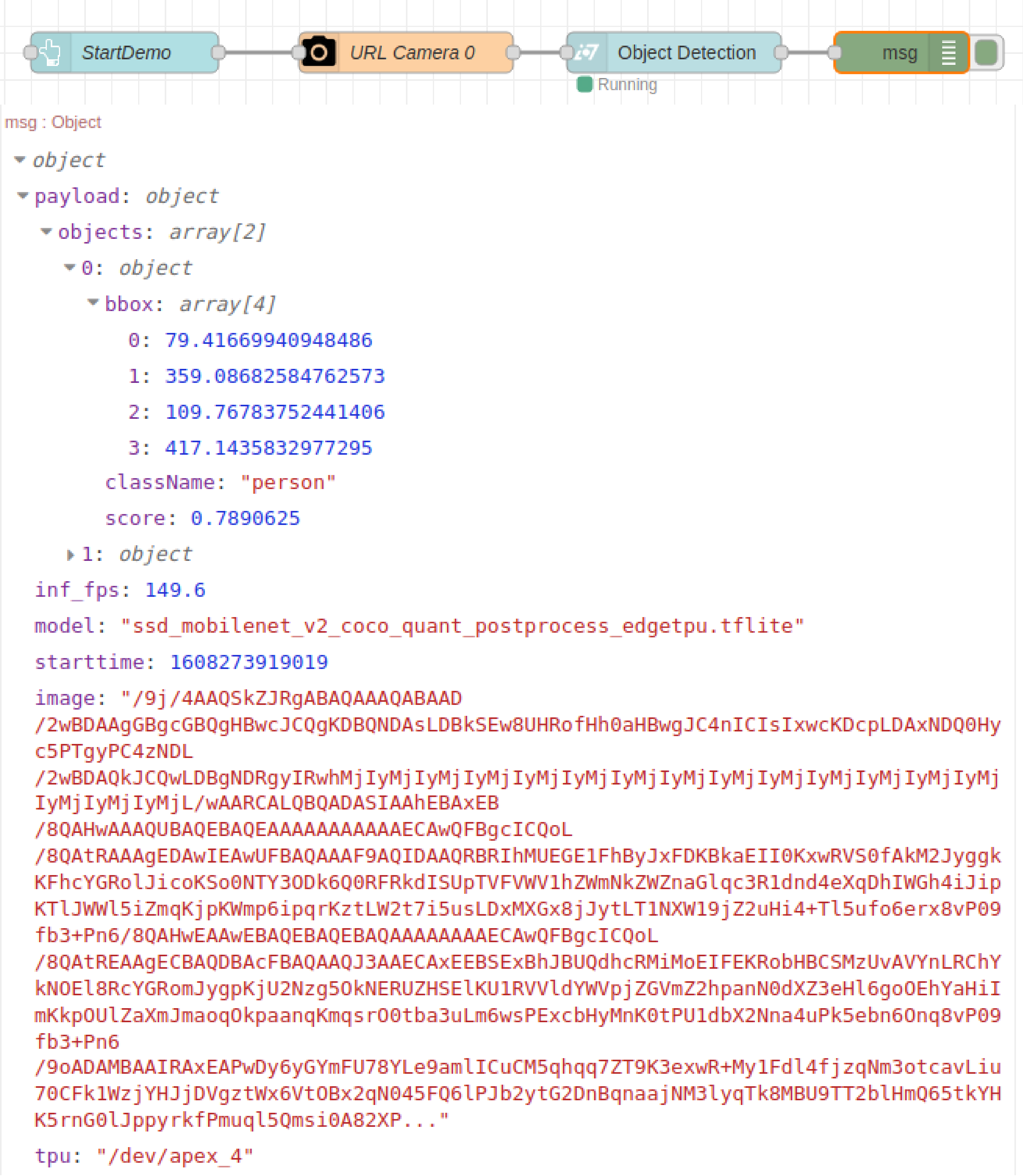


```
msg : Object
▾object
 ▾payload: object
   ▾classes: array[1]
     ▾0: object
        className: "container ship, containership, container vessel"
        score: 0.99609375
  inf_fps: 9.38
  model: "inception_v4_299_quant_edgetpu.tflite"
  starttime: 1608273300343
  image: "/9j/4AAQSkZJRgABAQAAAQABAAD
  /2wBDAAgGBgcGBQgHBwcJCQgKDBQNDAsLDBkSEw8UHRofHh0aHBwgJC4nICIsIxwcKDcpLDAxNDQ0Hy
  c5PTgyPC4zNDL
  /2wBDAQkJCQwLDBgNDRgyIRwhMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMj
  IyMjIyMjIyMjL/wAARCALQBQADASIAAhEBAxEB
  /8QAHwAAAQUBAQEBAQEAAAAAAAAAAAECAwQFBgcICQoL
  /8QAtRAAAgEDAwIEAwUFBAQAAAF9AQIDAAQRBRIhMUEGE1FhByJxFDKBkaEII0KxwRVS0fAkM2Jyggk
  KFhcYGRolJicoKSo0NTY3ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqDhIWGh4iJip
  KTlJWWl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uHi4+Tl5ufo6erx8vP09
  fb3+Pn6/8QAHwEAAwEBAQEBAQEBAQAAAAAAAAECAwQFBgcICQoL
  /8QAtREAAgECBAQDBAcFBAQAAQJ3AAECAxEEBSExBhJBUQdhcRMiMoEIFEKRobHBCSMzUvAVYnLRChY
  kNOEl8RcYGRomJygpKjU2Nzg5OkNERUZHSElKU1RVVldYWVpjZGVmZ2hpanN0dXZ3eHl6goOEhYaHiI
  mKkpOUlZaXmJmaoqOkpaanqKmqsrO0tba3uLm6wsPExcbHyMnK0tPU1dbX2Nna4uPk5ebn6Onq8vP09
  fb3+Pn6/9oADAMBAAIRAxEAPwDxFV3HApzRtGfmFaEFrArxSOzKjNgkdqmvNKuBGsqyx+RIu
  /g9u1bkFOOWLymBeMbUP3xV7Tri2Fqlu+ItoYsxXdk+31/pVNdLmSQMMGj7LNmTCcIu4
  /SggiWBpZxGgyzZx9aaFaGUhgQR1FWxbNbTRmZlUPHvDA5AHvUTrDHG..."
  tpu: "/dev/apex_7"
```

2.2 SZ Object Detection node output json format:

Output	**Format**	**Description**
bbox	array	The coordinate about x1, y1, x2 and y2 return from the edgetpu object detection api.
className	Strings	Class Name category
score	Integer	The percent about 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 transfer)
model	Strings	Inference node import the model’s filename
tpu	Strings	TPU node be used by inference node

Reference the Results on Node-red debug message:



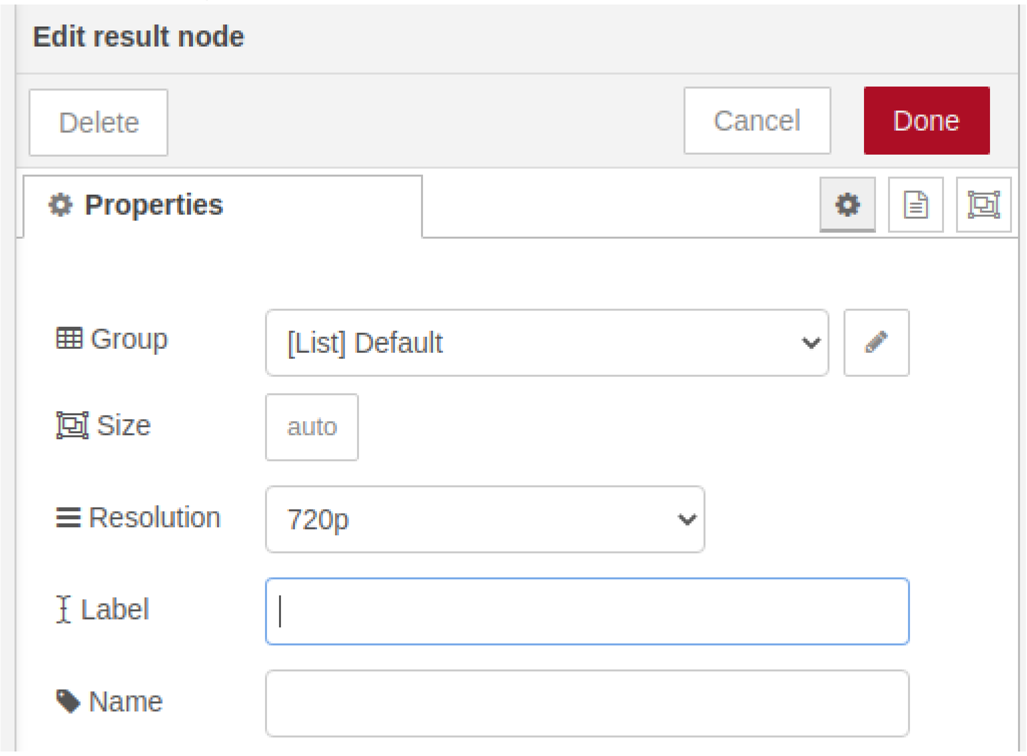
```
msg : Object
▾ object
 ▾payload: object
   ▾objects: array[2]
     ▾0: object
       ▾bbox: array[4]
           0: 79.41669940948486
           1: 359.08682584762573
           2: 109.76783752441406
           3: 417.1435832977295
         className: "person"
         score: 0.7890625
     ▸1: object
   inf_fps: 149.6
   model: "ssd_mobilenet_v2_coco_quant_postprocess_edgetpu.tflite"
   starttime: 1608273919019
   image: "/9j/4AAQSkZJRgABAQAAAQABAAD
   /2wBDAAgGBgcGBQgHBwcJCQgKDBQNDAsLDBkSEw8UHRofHh0aHBwgJC4nICIsIxwcKDcpLDAxNDQ0Hy
   c5PTgyPC4zNDL
   /2wBDAQkJCQwLDBgNDRgyIRwhMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMj
   IyMjIyMjIyMjL/wAARCALQBQADASIAAhEBAxEB
   /8QAHwAAAQUBAQEBAQEAAAAAAAAAAAECAwQFBgcICQoL
   /8QAtRAAAgEDAwIEAwUFBAQAAAF9AQIDAAQRBRIhMUEGE1FhByJxFDKBkaEII0KxwRVS0fAkM2Jyggk
   KFhcYGRolJicoKSo0NTY3ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqDhIWGh4iJip
   KTlJWWl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uHi4+Tl5ufo6erx8vP09
   fb3+Pn6/8QAHwEAAwEBAQEBAQEBAQAAAAAAAAECAwQFBgcICQoL
   /8QAtREAAgECBAQDBAcFBAQAAQJ3AAECAxEEBSExBhJBUQdhcRMiMoEIFEKRobHBCSMzUvAVYnLRChY
   kNOEl8RcYGRomJygpKjU2Nzg5OkNERUZHSElKU1RVVldYWVpjZGVmZ2hpanN0dXZ3eHl6goOEhYaHiI
   mKkpOUlZaXmJmaoqOkpaanqKmqsrO0tba3uLm6wsPExcbHyMnK0tPU1dbX2Nna4uPk5ebn6Onq8vP09
   fb3+Pn6
   /9oADAMBAAIRAxEAPwDy6yGYmFU78YLe9amlICuCM5qhqq7ZT9K3exwR+My1Fdl4fjzqNm3otcavLiu
   70CFk1WzjYHJjDVgztWx6VtOBx2qN045FQ6lPJb2ytG2DnBqnaajNM3lyqTk8MBU9TT2blHmQ65tkYH
   K5rnG0lJppyrkfPmuql5Qmsi0A82XP..."
   tpu: "/dev/apex_4"
```

- ## Result node

Show the output results from Inference Node:

- 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. Resolution：Reconfig the resolution of frames
4. Label：Show the topic on the dashboard
5. Name：Config and show the name on the node



Edit result node

Group [List] Default

Size auto

Resolution 720p

Label

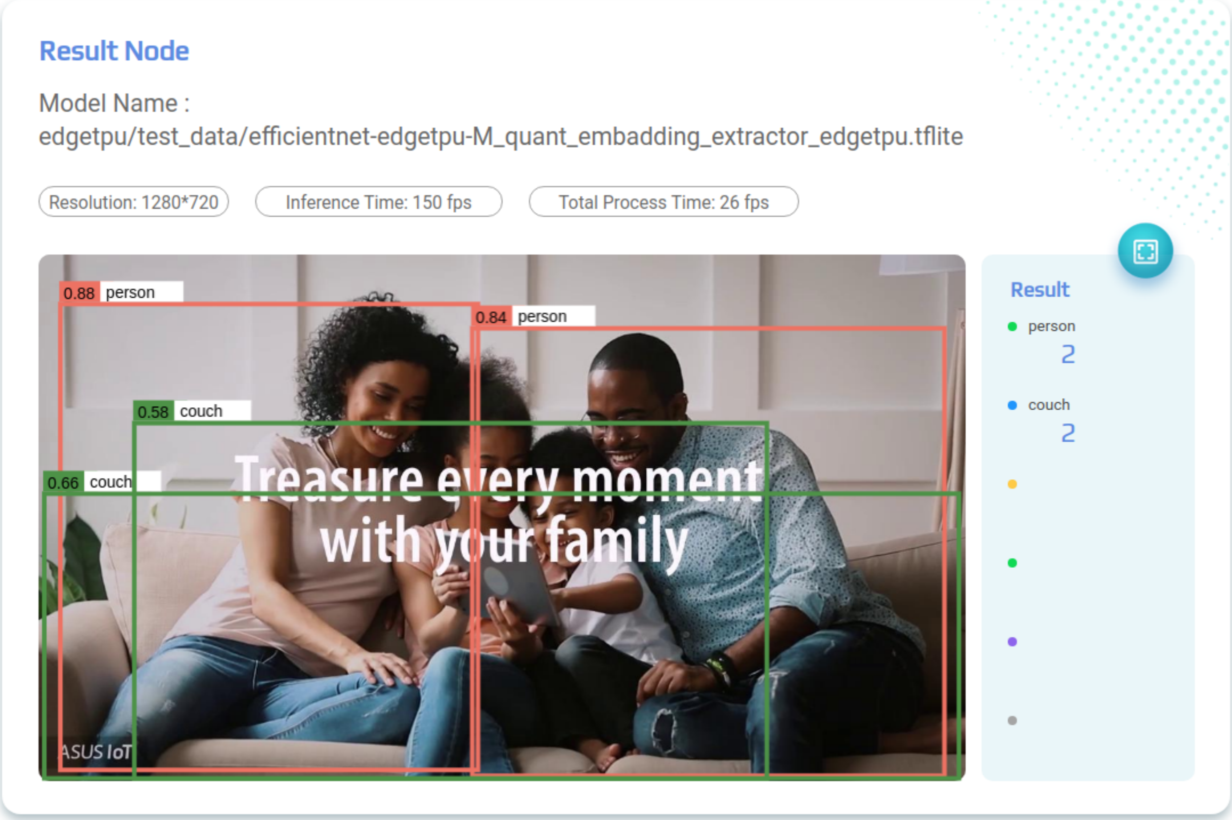
Name

- Output

Reference the following image about the node outputs:

1. Default mapping to Group
2. Size set “auto” and the video source is 720p
3. Label mapping to the Label config of properties
4. The colors of bounding-box are classified with people and not people:
   - people : Orange
   - not people : Blue

5. The right filed (Total): show the quantity of classification object



Result Node

Model Name :
edgetpu/test_data/efficientnet-edgetpu-M_quant_embadding_extractor_edgetpu.tflite

Resolution: 1280*720 | Inference Time: 150 fps | Total Process Time: 26 fps

Result
- person 2
- couch 2

- ## Performance node

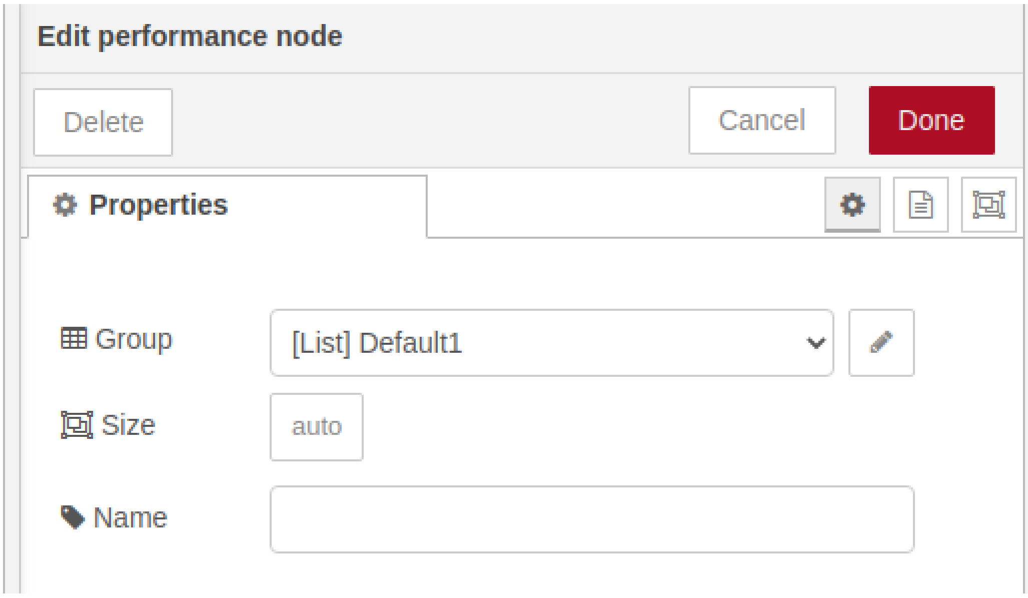
Show the following system information:

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. Name：Config and show the name on the node



Edit performance node

Group [List] Default1

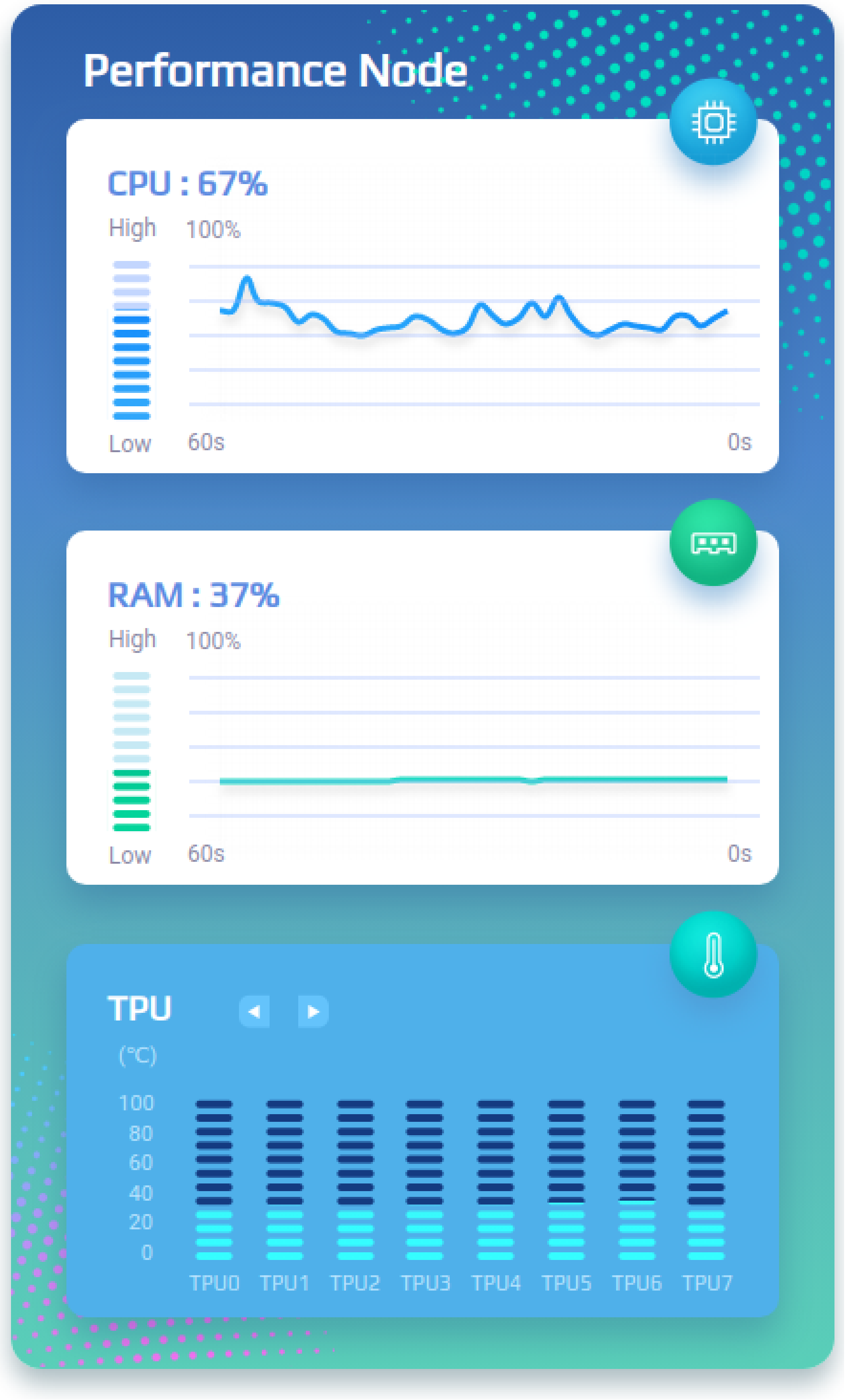
Size auto

Name

- **Output**

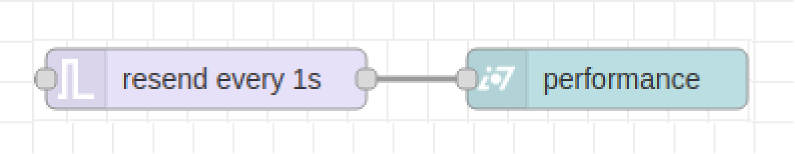
1. Default1 mapping to Group

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

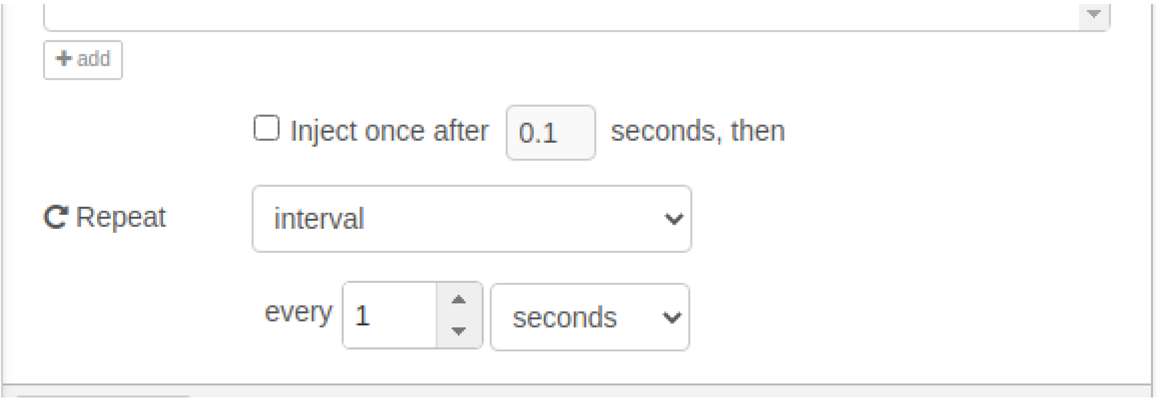


- Exmaple flow

Get a “inject node” and then connect to performance node:



Set the “Repeat” filed of “inject node” to “interval” every 1 seconds and then the performance node will update the result per second

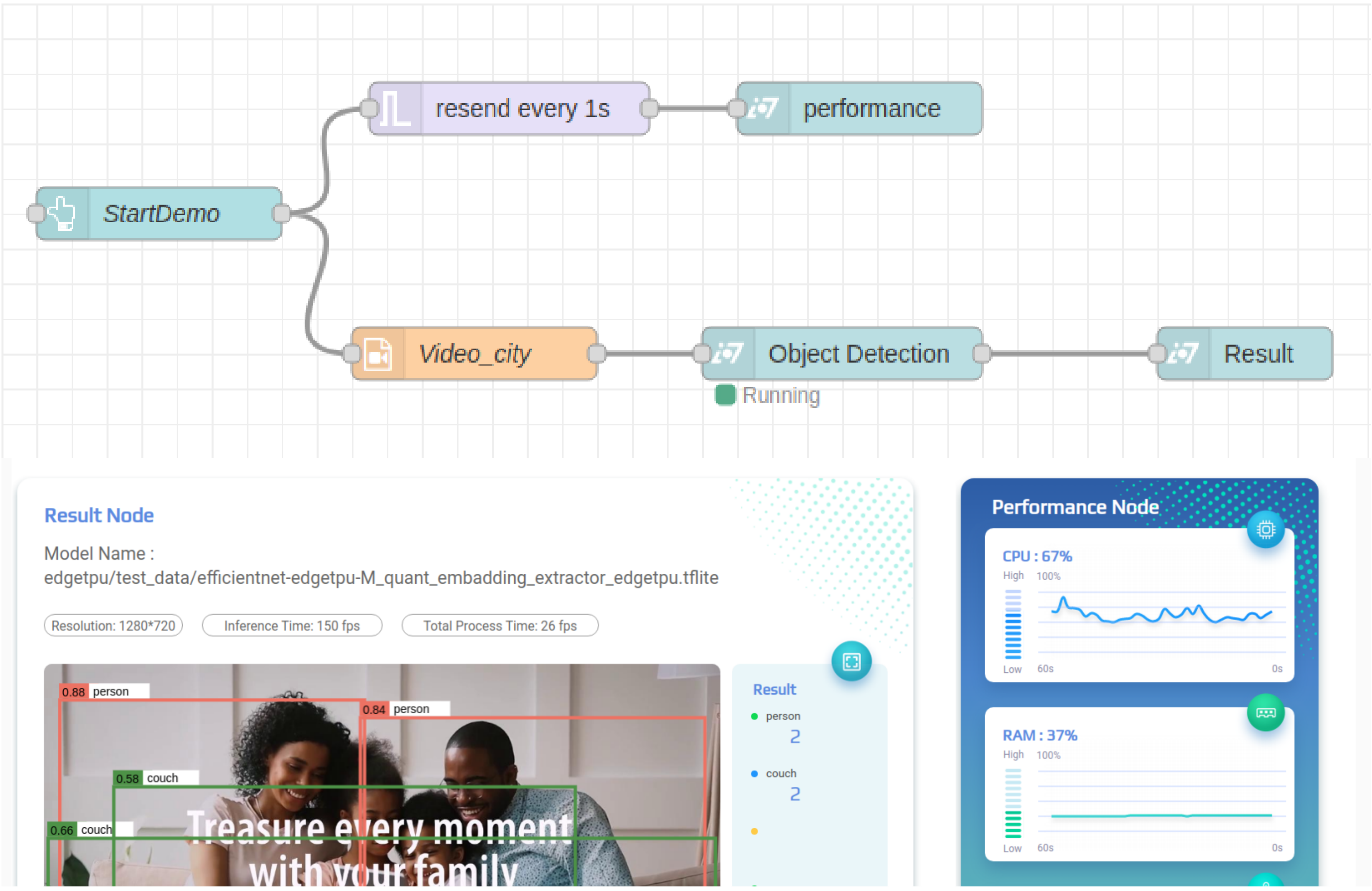


☐ Inject once after 0.1 seconds, then

Repeat: interval

every 1 seconds

node-red-contrib-edgetpu-inference Example Flow



Result Node

Model Name : edgetpu/test_data/efficientnet-edgetpu-M_quant_embadding_extractor_edgetpu.tflite

Resolution: 1280*720

Inference Time: 150 fps

Total Process Time: 26 fps

Result

- person 2
- couch 2

CPU : 67%

RAM : 37%

```
[
    {
        "id": "484a4066.57495",
        "type": "tab",
        "label": "result flow",
        "disabled": false,
        "info": ""
    },
    {
        "id": "b62b54bc.47f5b8",
        "type": "function",
        "z": "484a4066.57495",
        "name": "Video_city",
        "func": "\nmsg.payload=\"./node_modules/node-red-contrib-edgetpu-inference/tes
        "outputs": 1,
        "noerr": 0,
        "initialize": "",
        "finalize": "",
        "x": 330,
        "y": 460,
        "wires": [
            [
                "9ddba8c7.d7aa58"
            ]
        ],
        "icon": "font-awesome/fa-file-video-o"
    },
    {
        "id": "9ddba8c7.d7aa58",
        "type": "SZ Object Detection",
        "z": "484a4066.57495",
        "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/cocc
        "threshold": "0.5",
        "topk": "5",
        "keepratio": "0",
        "relativecoord": "0",
        "resample": "0",
        "outimage": true,
        "loop": true,
        "x": 540,
        "y": 460,
        "wires": [
            [
                "eb2c99f7.87bd78"
            ]
        ]
    },
    {
        "id": "eb2c99f7.87bd78",
        "type": "ui_result",
        "z": "484a4066.57495",
        "group": "365bb82b.fd8d18",
        "name": "",
        "title": "Label",
        "order": 1,
        "resolution": "1",
        "width": "21",
        "height": "14",
        "x": 770,
        "y": 460,
        "wires": []
    },
    {
        "id": "d3960755.7f39d8",
        "type": "ui_performance",
        "z": "484a4066.57495",
        "group": "365bb82b.fd8d18",
        "name": "",
        "title": "",
        "order": 2,
        "width": "9",
        "height": "14",
        "x": 550,
        "y": 300,
        "wires": []
    },
    {
        "id": "24bc2238.e885be",
        "type": "ui_button",
        "z": "484a4066.57495",
        "name": "StartDemo",
        "group": "365bb82b.fd8d18",
        "order": 1,
        "width": 0,
        "height": 0,
        "passthru": false,
        "label": "StartDemo",
        "tooltip": "",
        "color": "",
        "bgcolor": "",
        "icon": "",
        "payload": "",
        "payloadType": "str",
        "topic": "",
        "x": 150,
        "y": 380,
        "wires": [
            [
                "b62b54bc.47f5b8",
                "44161e7.d23cbe"
            ]
        ]
    },
    {
        "id": "44161e7.d23cbe",
        "type": "trigger",
        "z": "484a4066.57495",
        "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": [
            [
                "d3960755.7f39d8"
            ]
        ]
    },
    {
        "id": "365bb82b.fd8d18",
        "type": "ui_group",
        "z": "",
        "name": "Default1",
        "tab": "36b75a6f.51b2c6",
        "order": 3,
        "disp": true,
        "width": "30",
        "collapse": false
    },
    {
        "id": "36b75a6f.51b2c6",
        "type": "ui_tab",
        "z": "",
        "name": "List",
        "icon": "dashboard"
    }
]
```