

node-red-contrib-edge-tpu 0.0.1

Node Red nodes to inference AI models, show inference results 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-edge-tpu
```

Category

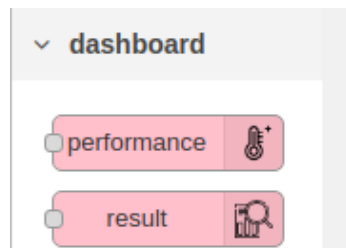
- Inference Node

Find the Inference Node from the following "Shen Zhou" category:



- Result node and Performance node

Find the Inference Node from the following "dashboard" category:



Usage

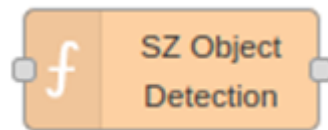
• Inference Node

> “SZ Image Classification” node:



Run the image classification models by TPUs

> “SZ Object Detection” node:



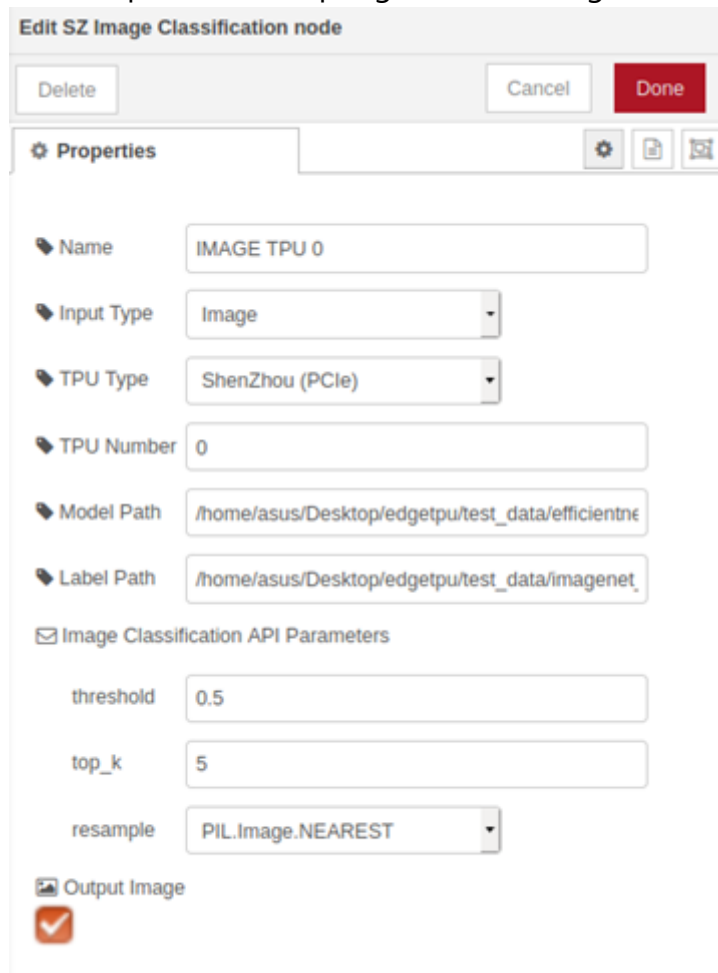
Run the Object detection models by TPUs

- Node Properties:

The explanation about the items of Properties:

1. Input Type : The source input format. Include 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 Number : Select a Number for 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 returned classifications.
 - o top_k : The maximum number of classifications to return.

o resample : A resampling filter for image resizing.



Edit SZ Image Classification node

Delete Cancel Done

Properties

Name IMAGE TPU 0

Input Type Image

TPU Type ShenZhou (PCIe)

TPU Number 0

Model Path /home/asus/Desktop/edgetpu/test_data/efficientne

Label Path /home/asus/Desktop/edgetpu/test_data/imagenet

Image Classification API Parameters

threshold 0.5

top_k 5

resample PIL.Image.NEAREST

Output Image

☒

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, keep the image aspect ratio the same when down-sampling the image. If False, resize and reshape the image (without cropping) to match the input tensor's dimensions.

o relative_coord : If True, provide coordinates as float values between 0 and 1, representing each position relative to the total image width/height. If False, provide 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 SZ Object Detection node

Delete Cancel Done

Properties

Name: Object TPU 3

Input Type: URL

TPU Type: ShenZhou (PCIe)

TPU Number: 3

Model Path: /home/asus/Desktop/edgetpu/test_data/ssd_mobi

Label Path: /home/asus/Desktop/edgetpu/test_data/coco_label

Object Detection API Parameters

threshold: 0.1

top_k: 4

keep_ratio: False

relative_coord: False

resample: PIL.Image.NEAREST

Output Image ☒

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>)

- Input and Output data Formats:

1. Input data format to inference node:

When performing inference using a edgetpu model, 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%E2%80%9D)
Video	Strings	"/home/asus/Desktop/test.mp4"
Local Camera	Strings	"0"

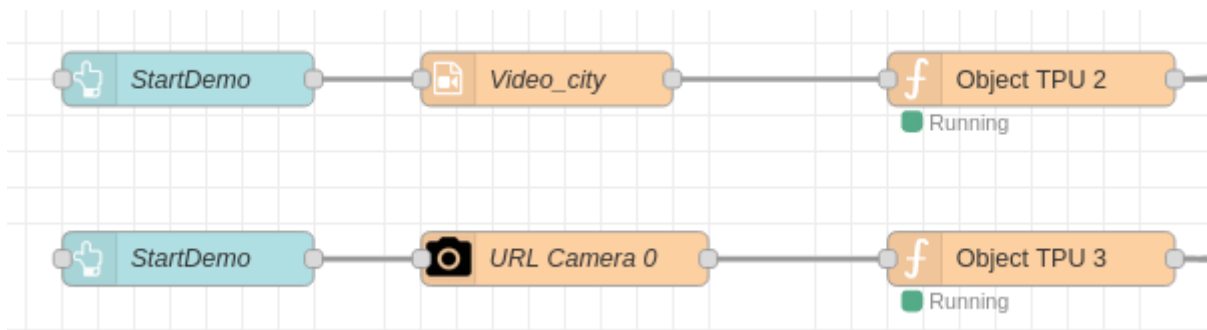
Example input to inference node when the source is a url streaming server source:

```
{
  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:

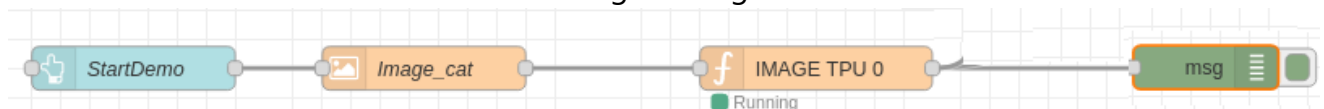


2. Output data format from inference node::

2.1 SZ Image Classification node output json format:

Output	Format	Description
className	Strings	Class Name category
score	Integer	The 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)

Reference the Results on Node-red debug message:



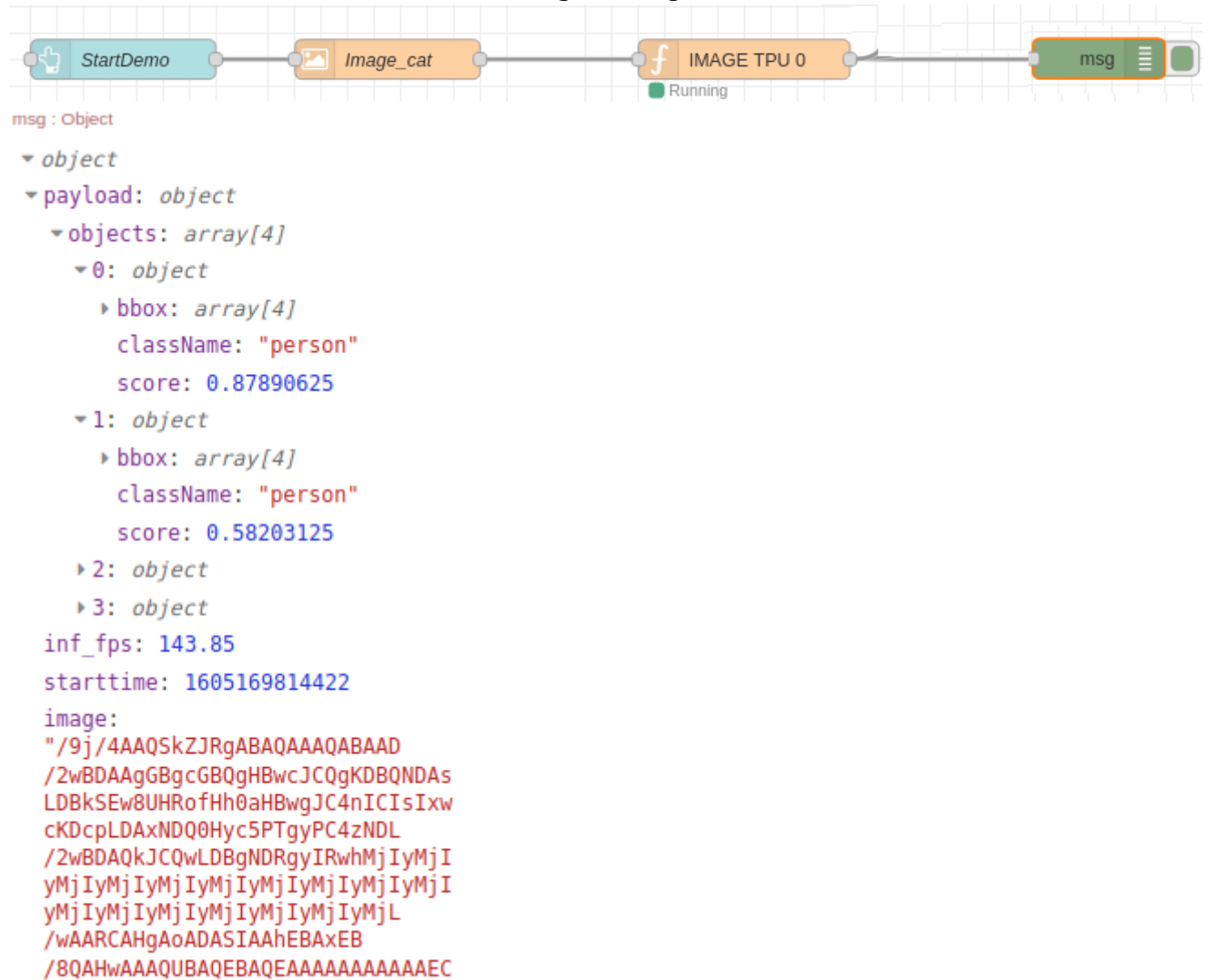
```

msg : Object
  ▾ object
    ▾ payload: object
      ▾ classes: array[1]
        ▾ 0: object
          className: "chickadee"
          score: 0.640625
        inf_fps: 32.2
        starttime: 1605168875943
        image:
          "/9j/4AAQSkZJRgABAQAAQABAAQ
          /2wBDAAgGBgcGBQgHBwcJCQgKDBQNDAs
          LDBkSEw8UHRofHh0aHBwgJC4nICIsIxw
          cKDcpLDAxNDQ0Hyc5PTgyPC4zNDL
          /2wBDAQkJCQwLDBgNDRgyIRwhMjIyMjI
          yMjIyMjIyMjIyMjIyMjIyMjIyMjI
          yMjIyMjIyMjIyMjIyMjIyMjIyMjL
          /wAARCAFXAgADASIAAhEBAxEB
          /8QAHwAAAQUBAQEBAQEAAAAAAAAAAAE
  
```

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)

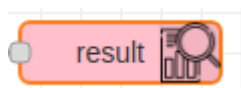
Reference the Results on Node-red debug message:



The image shows a Node-RED flow diagram and its corresponding debug console output. The flow consists of four nodes: 'StartDemo' (button), 'Image_cat' (image input), 'IMAGE TPU 0' (function node, currently running), and 'msg' (message output). The debug console displays the following message:

```
msg : Object
  object
  payload: object
    objects: array[4]
      0: object
        bbox: array[4]
        className: "person"
        score: 0.87890625
      1: object
        bbox: array[4]
        className: "person"
        score: 0.58203125
      2: object
      3: object
    inf_fps: 143.85
    starttime: 1605169814422
    image:
      "/9j/4AAQSkZJRgABAQAAQABAAD
      /2wBDAAGGBgcGBQgHBwcJCQgKDBQNDAs
      LDBkSEw8UHRofHh0aHBwgJC4nICIsIxw
      cKDcpLDAxNDQ0Hyc5PTgyPC4zNDL
      /2wBDAQkJCQwLDBgNDRgyIRwhMjIyMjI
      yMjIyMjIyMjIyMjIyMjIyMjIyMjI
      yMjIyMjIyMjIyMjIyMjIyMjL
      /wAARCAHgAoADASIAAhEBAxEB
      /8QAHwAAAQUBAQEBAQEAAAAAAAAAAAEc
```

• 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

Delete Cancel Done

Properties

Group [List] Default

Size auto

Resolution 720p

Label I

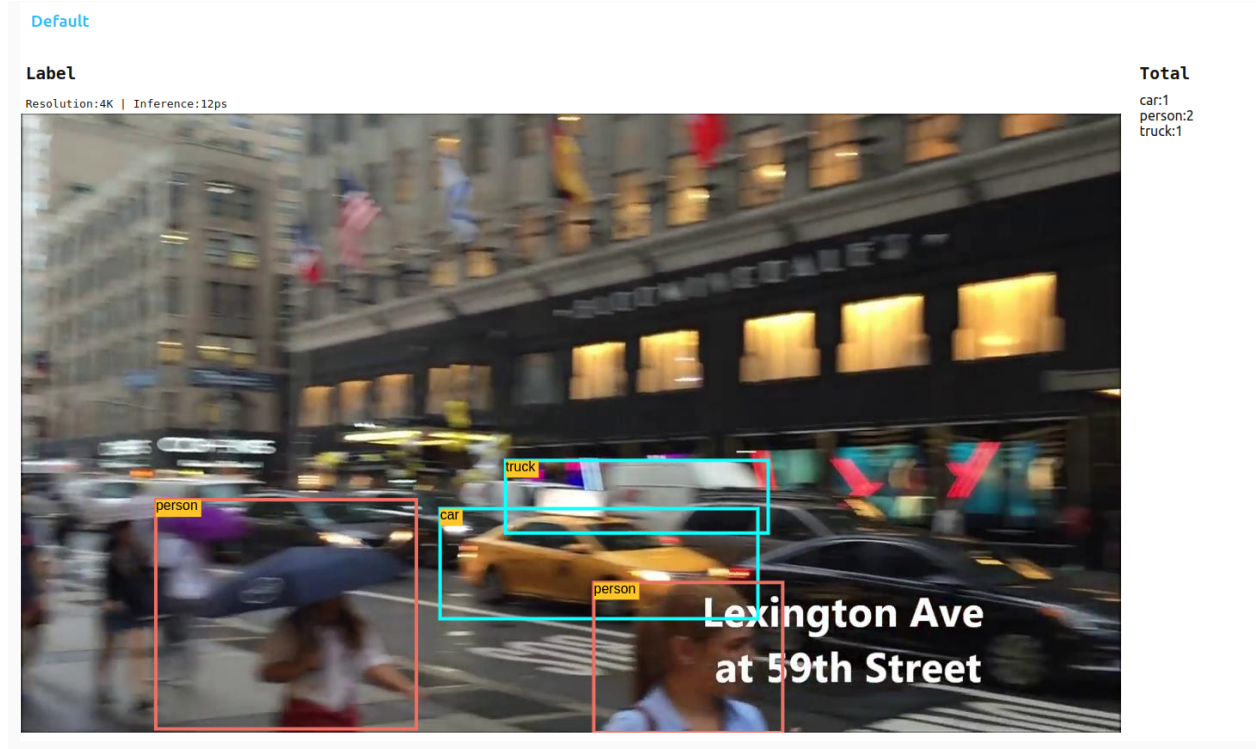
Name

- Output

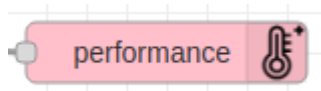
Reference the following image about the node outputs:

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



• 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

Delete Cancel Done

Properties

Group [List] Default1

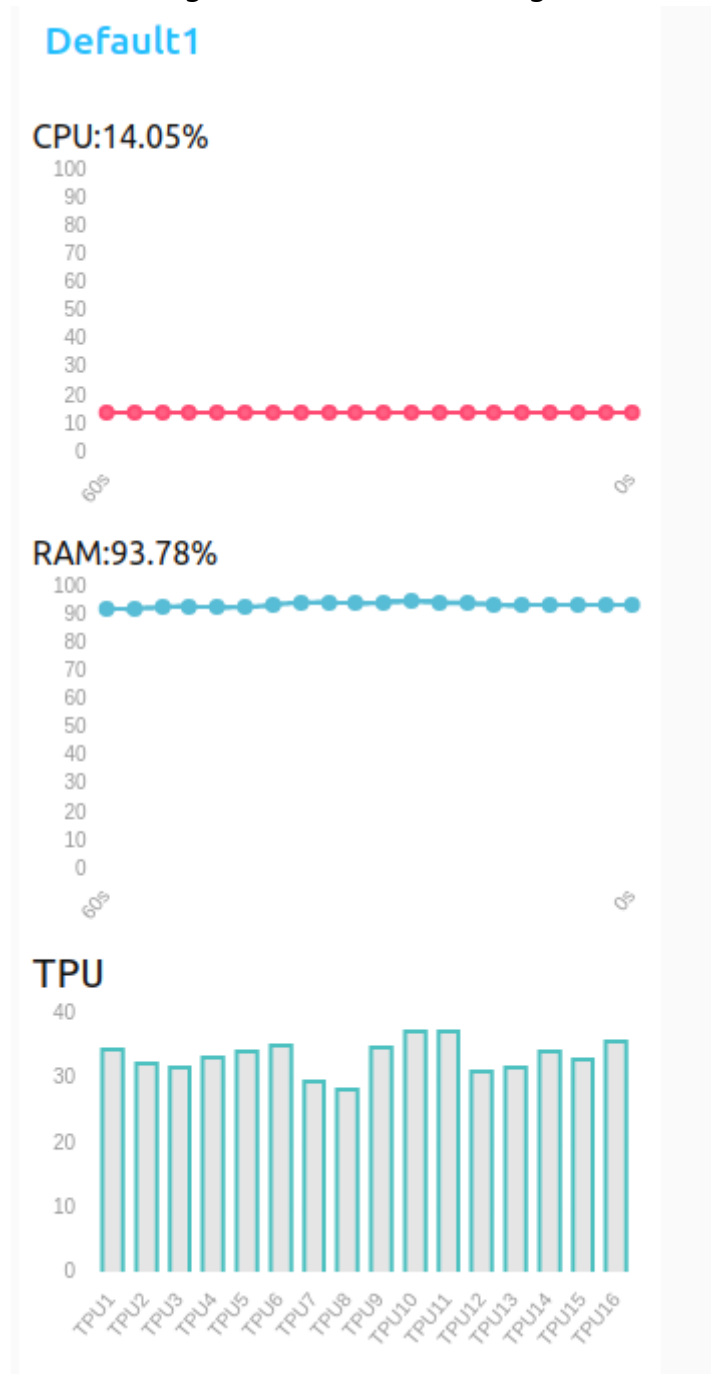
Size auto

Name

- Output

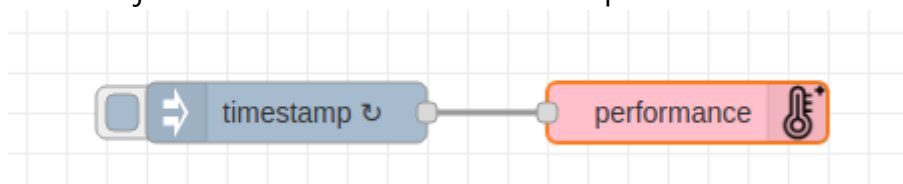
1. Default1 mapping to Group

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

+ add

☐ Inject once after

0.1

 seconds, then

Repeat

interval

▼

every

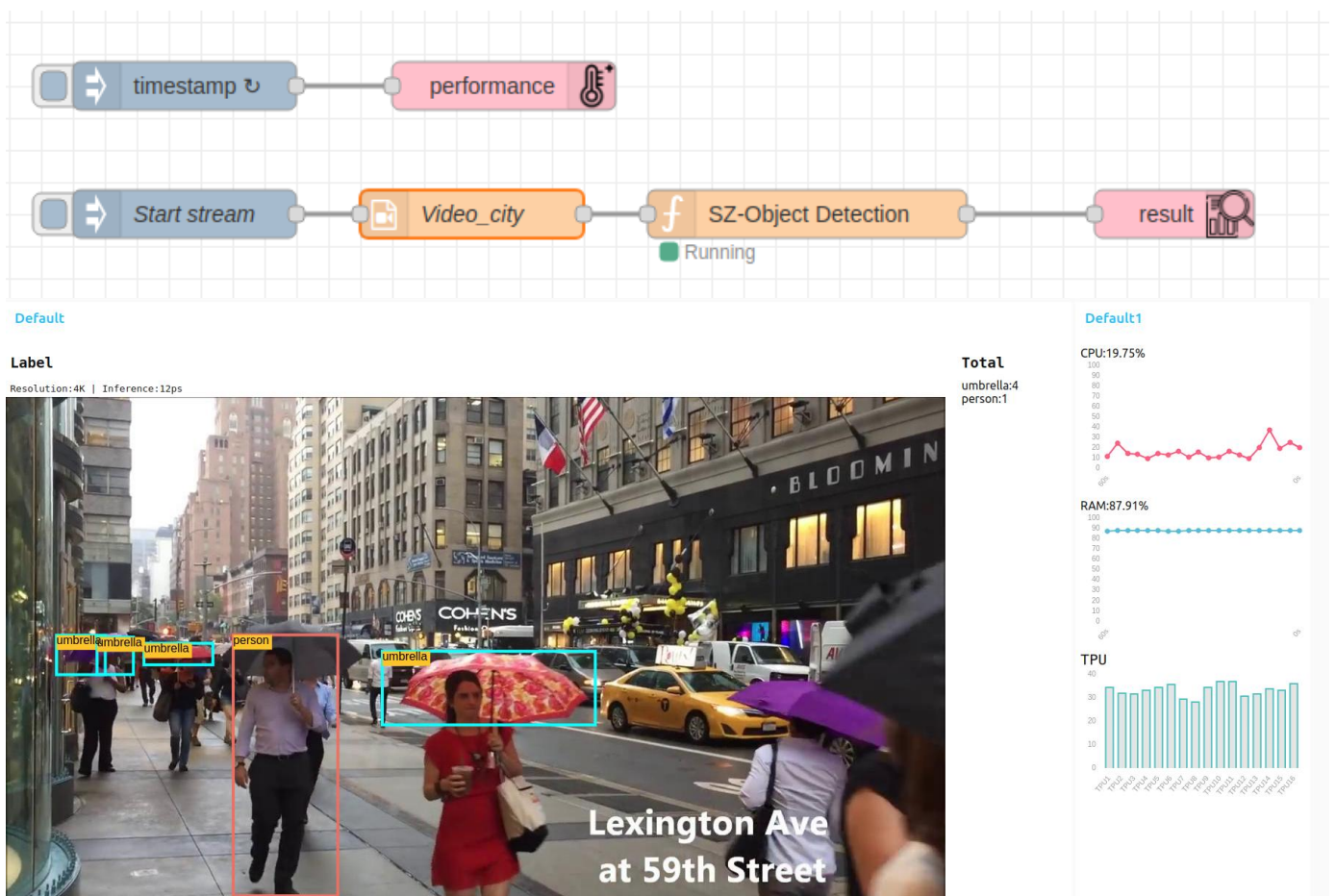
1

▼

seconds

▼

node-red-contrib-edge-tpu Example Flow



```
[
  {
    "id": "b5cf4e81.1b595",
    "type": "tab",
    "label": "result flow",
    "disabled": false,
    "info": ""
  },
  {
    "id": "d7abc15d.9c02b",
    "type": "function",
    "z": "b5cf4e81.1b595",
    "name": "Video_city",
    "func": "msg.payload=\"/home/asus/Downloads/test.mp4\"\\nreturn msg;",
    "outputs": 1,
    "noerr": 0,
    "initialize": "",
    "finalize": "",
    "x": 310,
    "y": 460,
    "wires": [
      [
        "2013f5c5.9da0ea"
      ]
    ],
    "icon": "font-awesome/fa-file-video-o"
  },
  {
    "id": "55857370.df40ec",
    "type": "inject",
    "z": "b5cf4e81.1b595",
    "name": "Start stream",
    "props": [
      {
        "p": "payload"
      },
      {
        "p": "topic",
        "vt": "str"
      }
    ],
    "repeat": "",
    "crontab": "",
    "once": false,
    "onceDelay": "",
    "topic": "",
    "payload": "",
    "payloadType": "date",
    "x": 130,
    "y": 460,
    "wires": [
      [
        "d7abc15d.9c02b"
      ]
    ]
  }
]
```

```

    ]
  },
  {
    "id": "2013f5c5.9da0ea",
    "type": "SZ Object Detection",
    "z": "b5cf4e81.1b595",
    "name": "",
    "intype": "2",
    "tputype": "0",
    "tpunum": "1",
    "modelpath": "/home/asus/Desktop/edgetpu/test_data/ssd_mobilenet_v1_coco_quant",
    "labelpath": "/home/asus/Desktop/edgetpu/test_data/coco_labels.txt",
    "threshold": "0.5",
    "topk": "5",
    "keepratio": "0",
    "relativecoord": "0",
    "resample": "0",
    "outimage": true,
    "x": 520,
    "y": 460,
    "wires": [
      [
        "9f0aa671.73e948"
      ]
    ]
  },
  {
    "id": "d4d2ff52.326cd",
    "type": "inject",
    "z": "b5cf4e81.1b595",
    "name": "",
    "props": [
      {
        "p": "payload"
      },
      {
        "p": "topic",
        "vt": "str"
      }
    ],
    "repeat": "1",
    "crontab": "",
    "once": false,
    "onceDelay": 0.1,
    "topic": "",
    "payload": "",
    "payloadType": "date",
    "x": 110,
    "y": 320,
    "wires": [
      [
        "da450bda.80ee18"
      ]
    ]
  },
  },

```

```

{
  "id": "9f0aa671.73e948",
  "type": "ui_result",
  "z": "b5cf4e81.1b595",
  "group": "cd8d5087.f050d",
  "name": "",
  "title": "Label",
  "order": 0,
  "resolution": "1",
  "width": "0",
  "height": "0",
  "x": 750,
  "y": 460,
  "wires": []
},
{
  "id": "da450bda.80ee18",
  "type": "ui_performance",
  "z": "b5cf4e81.1b595",
  "group": "79ffb604.012868",
  "name": "",
  "title": "",
  "order": 0,
  "width": 0,
  "height": 0,
  "x": 330,
  "y": 320,
  "wires": []
},
{
  "id": "cd8d5087.f050d",
  "type": "ui_group",
  "z": "",
  "name": "Default",
  "tab": "36b75a6f.51b2c6",
  "order": 2,
  "disp": true,
  "width": "27",
  "collapse": false
},
{
  "id": "79ffb604.012868",
  "type": "ui_group",
  "z": "",
  "name": "Default1",
  "tab": "36b75a6f.51b2c6",
  "order": 3,
  "disp": true,
  "width": "6",
  "collapse": false
},
{
  "id": "36b75a6f.51b2c6",
  "type": "ui_tab",
  "z": "",

```

```
    "name": "List",  
    "icon": "dashboard"  
  }  
]
```

