# `recognition` : \*string\*

    识别算法类型。可选，默认 `DirectHit` 。

    可选的值：`DirectHit` | `TemplateMatch` | `FeatureMatch` | `ColorMatch` | `OCR` | `NeuralNetworkClassify` | `NeuralNetworkDetect` | `Custom`

算法类型

## ### `DirectHit`

直接命中，即不进行识别，直接执行动作。

## ### `TemplateMatch`

模板匹配，即“找图”。

该算法属性需额外部分字段：

- `roi`: \*array<int, 4>\* | \*string\*

    识别区域坐标。可选，默认 [0, 0, 0, 0] ，即全屏。

  - \*array<int, 4>\*: 识别区域坐标，[x, y, w, h]，若希望全屏可设为 [0, 0, 0, 0] 。

  - \*string\*: 填写节点名，在之前执行过的某节点识别到的目标范围内识别。

- `roi\_offset`: \*array<int, 4>\*

    在 `roi` 的基础上额外移动再作为范围，四个值分别相加。可选，默认 [0, 0, 0, 0] 。

- `template`: \*string\* | \*list<string, >\*

    模板图片路径，需要 `image` 文件夹的相对路径。必选。

    所使用的图片需要是无损原图缩放到 720p 后的裁剪。请参考 [这里](1.1-快速开始.md#图片文件)。

- `threshold`: \*double\* | \*list<double, >\*

    模板匹配阈值。可选，默认 0.7 。

    若为数组，长度需和 `template` 数组长度相同。

- `order\_by`: \*string\*

    结果排序方式。可选，默认 `Horizontal`。

    可选的值：`Horizontal` | `Vertical` | `Score` | `Random` 。

    可结合 `index` 字段使用。

- `index`: \*int\*

    命中第几个结果。可选，默认 0 。

    假设共有 N 个结果，则 `index` 的取值范围为 [-N, N - 1] ，其中负数使用类 Python 的规则转换为 N - index 。若超出范围，则视为当前识别无结果。

- `method`: \*int\*

    模板匹配算法，即 cv::TemplateMatchModes。可选，默认 5 。

    仅支持 1、3、5，可简单理解为越大的越精确，但也会更慢。

    详情请参考 [OpenCV 官方文档](https://docs.opencv.org/4.x/df/dfb/group\_\_imgproc\_\_object.html)。

- `green\_mask`: \*bool\*

    是否进行绿色掩码。可选，默认 false 。

    若为 true，可以将图片中不希望匹配的部分涂绿 RGB: (0, 255, 0)，则不对绿色部分进行匹配。

## ### `FeatureMatch`

特征匹配，泛化能力更强的“找图”，具有抗透视、抗尺寸变化等特点。

该算法属性需额外部分字段：

- `roi`: \*array<int, 4>\* | \*string\*

    同 `TemplateMatch`.`roi` 。

- `roi\_offset`: \*array<int, 4>\*

    同 `TemplateMatch`.`roi\_offset` 。

- `template`: \*string\* | \*list<string, >\*

    模板图片路径，需要 `image` 文件夹的相对路径。必选。

- `count`: \*int\*

    匹配的特征点的数量要求（阈值），默认 4 。

- `order\_by`: \*string\*

    结果排序方式。可选，默认 `Horizontal` 。

    可选的值：`Horizontal` | `Vertical` | `Score` | `Area` | `Random` 。

    可结合 `index` 字段使用。

- `index`: \*int\*

    命中第几个结果。可选，默认 0 。

    假设共有 N 个结果，则 `index` 的取值范围为 [-N, N - 1] ，其中负数使用类 Python 的规则转换为 N - index 。若超出范围，则视为当前识别无结果。

- `green\_mask`: \*bool\*

    是否进行绿色掩码。可选，默认 false 。

    若为 true，可以将图片中不希望匹配的部分涂绿 RGB: (0, 255, 0)，则不对绿色部分进行匹配。

- `detector`: \*string\*

    特征检测器。可选，默认 `SIFT` 。

    目前支持以下算法：

  - SIFT

    计算复杂度高，具有尺度不变性、旋转不变性。效果最好。

  - KAZE

    适用于2D和3D图像，具有尺度不变性、旋转不变性。

  - AKAZE

    计算速度较快，具有尺度不变性、旋转不变性。

  - BRISK

    计算速度非常快，具有尺度不变性、旋转不变性。

  - ORB

    计算速度非常快，具有旋转不变性。但不具有尺度不变性。

  各算法特点详情可自行进一步查询。

- `ratio`: \*double\*

    KNN 匹配算法的距离比值，[0 - 1.0] , 越大则匹配越宽松（更容易连线）。可选，默认 0.6 。

## ### `ColorMatch`

颜色匹配，即“找色”。

该算法属性需额外部分字段：

- `roi`: \*array<int, 4>\* | \*string\*

    同 `TemplateMatch`.`roi` 。

- `roi\_offset`: \*array<int, 4>\*

    同 `TemplateMatch`.`roi\_offset` 。

- `method`: \*int\*

    颜色匹配方式。即 cv::ColorConversionCodes。可选，默认 4 (RGB) 。

    常用值：4 (RGB, 3 通道), 40 (HSV, 3 通道), 6 (GRAY, 1 通道)。

    详情请参考 [OpenCV 官方文档](https://docs.opencv.org/4.x/d8/d01/group\_\_imgproc\_\_color\_\_conversions.html)。

- `lower`: \*list<int, >\* | \*list<list<int, >>\*

    颜色下限值。必选。最内层 list 长度需和 `method` 的通道数一致。

- `upper`:  \*list<int, >\* | \*list<list<int, >>\*

    颜色上限值。必选。最内层 list 长度需和 `method` 的通道数一致。

- `count`: \*int\*

    符合的点的数量要求（阈值）。可选，默认 1。

- `order\_by`: \*string\*

    结果排序方式。可选，默认 `Horizontal` 。

    可选的值：`Horizontal` | `Vertical` | `Score` | `Area` | `Random` 。

    可结合 `index` 字段使用。

- `index`: \*int\*

    命中第几个结果。可选，默认 0 。

    假设共有 N 个结果，则 `index` 的取值范围为 [-N, N - 1] ，其中负数使用类 Python 的规则转换为 N - index 。若超出范围，则视为当前识别无结果。

- `connected`: \*bool\*

    是否是相连的点才会被计数。可选，默认 false 。

    若为是，在完成颜色过滤后，则只会计数像素点 \*\*全部相连\*\* 的最大块。

    若为否，则不考虑这些像素点是否相连。

## ### `OCR`

文字识别。

该算法属性需额外部分字段：

- `roi`: \*array<int, 4>\* | \*string\*

    同 `TemplateMatch`.`roi` 。

- `roi\_offset`: \*array<int, 4>\*

    同 `TemplateMatch`.`roi\_offset` 。

- `expected`: \*string\* | \*list<string, >\*

    期望的结果，支持正则。必选。

- `threshold`: \*double\*

    模型置信度阈值。可选，默认 0.3 。

- `replace`: \*array<string, 2>\* | \*list<array<string, 2>>\*

    部分文字识别结果不准确，进行替换。可选。

- `order\_by`: \*string\*

    结果排序方式。可选，默认 `Horizontal`。

    可选的值：`Horizontal` | `Vertical` | `Area` | `Length` | `Random` 。

    可结合 `index` 字段使用。

- `index`: \*int\*

    命中第几个结果。可选，默认 0 。

    假设共有 N 个结果，则 `index` 的取值范围为 [-N, N - 1] ，其中负数使用类 Python 的规则转换为 N - index 。若超出范围，则视为当前识别无结果。

- `only\_rec`: \*bool\*

    是否仅识别（不进行检测，需要精确设置 `roi`）。可选，默认 false 。

- `model`: \*string\*

    模型 \*\*文件夹\*\* 路径。使用 `model/ocr` 文件夹的相对路径。可选，默认为空。

    若为空，则为 `model/ocr` 根目录下的模型文件。

    文件夹中需要包含 `rec.onnx`, `det.onnx`, `keys.txt` 三个文件。

## ### `NeuralNetworkClassify`

深度学习分类，判断图像中的 \*\*固定位置\*\* 是否为预期的“类别”。

该算法属性需额外部分字段：

- `roi`: \*array<int, 4>\* | \*string\*

    同 `TemplateMatch`.`roi` 。

- `roi\_offset`: \*array<int, 4>\*

    同 `TemplateMatch`.`roi\_offset` 。

- `labels`: \*list<string, >\*

    标注，即每个分类的名字。可选。

    仅影响调试图片及日志等，若未填写则会填充 "Unknown" 。

- `model`: \*string\*

    模型文件路径。使用 `model/classify` 文件夹的相对路径。必选。

    目前仅支持 ONNX 模型，参考 [NNClassify 食谱](https://github.com/MaaXYZ/MaaNeuralNetworkCookbook/tree/main/NeuralNetworkClassify)。

- `expected`: \*int\* | \*list<int, >\*

    期望的分类下标。

- `order\_by`: \*string\*

    结果排序方式。可选，默认 `Horizontal` 。

    可选的值：`Horizontal` | `Vertical` | `Score` | `Random` 。

    可结合 `index` 字段使用。

- `index`: \*int\*

    命中第几个结果。可选，默认 0 。

    假设共有 N 个结果，则 `index` 的取值范围为 [-N, N - 1] ，其中负数使用类 Python 的规则转换为 N - index 。若超出范围，则视为当前识别无结果。

举例：例如画面中 \*\*固定位置\*\* 可能出现 猫、狗、老鼠，我们训练了支持该三分类的模型。

希望识别到 猫 或 老鼠 才点击，而识别到 狗 不点击，则相关字段为：

```jsonc

{

    "labels": ["Cat", "Dog", "Mouse"],

    "expected": [0, 2]

}

```

注意这些值需要与模型实际输出相符。

## ### `NeuralNetworkDetect`

深度学习目标检测，高级版“找图”。

与分类器主要区别在于“找”，即支持任意位置。但通常来说模型复杂度会更高，需要更多的训练集、训练时间，使用时的资源占用（推理开销）也会成倍上涨。

该算法属性需额外部分字段：

- `roi`: \*array<int, 4>\* | \*string\*

    同 `TemplateMatch`.`roi` 。

- `roi\_offset`: \*array<int, 4>\*

    同 `TemplateMatch`.`roi\_offset` 。

- `labels`: \*list<string, >\*

    标注，即每个分类的名字。可选。

    仅影响调试图片及日志等，若未填写则会填充 "Unknown" 。

- `model`: \*string\*

    模型文件路径。使用 `model/detect` 文件夹的相对路径。必选。

    目前支持 YoloV8 ONNX 模型，其他同样输入输出的 Yolo 模型理论上也可以支持，但未经测试。

    训练参考 [NNDetect 食谱](https://github.com/MaaXYZ/MaaNeuralNetworkCookbook/tree/main/NeuralNetworkDetect)。

- `expected`: \*int\* | \*list<int, >\*

    期望的分类下标。

- `threshold`: \*double\* | \*list<double, >\*

    模型置信度阈值。可选，默认 0.3 。

    若为数组，长度需和 `expected` 数组长度相同。

- `order\_by`: \*string\*

    结果排序方式。可选，默认 `Horizontal` 。

    可选的值：`Horizontal` | `Vertical` | `Score` | `Area` | `Random` 。

    可结合 `index` 字段使用。

- `index`: \*int\*

    命中第几个结果。可选，默认 0 。

    假设共有 N 个结果，则 `index` 的取值范围为 [-N, N - 1] ，其中负数使用类 Python 的规则转换为 N - index 。若超出范围，则视为当前识别无结果。

举例：例如画面中可能出现 猫、狗、老鼠，我们训练了支持该三分类的检测模型。

希望检测到 猫 或 老鼠 才点击，而识别到 狗 不点击，则相关字段为：

```jsonc

{

    "labels": ["Cat", "Dog", "Mouse"],

    "expected": [0, 2]

}

```

注意这些值需要与模型实际输出相符。

## ### `Custom`

执行通过 `MaaResourceRegisterCustomRecognition` 接口传入的识别器句柄。

该算法属性需额外部分字段：

- `custom\_recognition`: \*string\*

    识别名，同注册接口传入的识别名。同时会通过 `MaaCustomRecognitionCallback`.`custom\_recognition\_name` 传出。必选。

- `custom\_recognition\_param`: \*any\*

    识别参数，任意类型，会通过 `MaaCustomRecognitionCallback`.`custom\_recognition\_param` 传出。可选，默认空 json，即 `{}` 。

- `roi`: \*array<int, 4>\* | \*string\*

    同 `TemplateMatch`.`roi`，会通过 `MaaCustomRecognitionCallback`.`roi` 传出。可选，默认 [0, 0, 0, 0] 。

- `roi\_offset`: \*array<int, 4>\*

    同 `TemplateMatch`.`roi\_offset` 。

# ## 动作类型

## ### `DoNothing`

什么都不做。

## ### `Click`

点击。

该动作属性需额外部分字段：

- `target`: \*true\* | \*string\* | \*array<int, 4>\*

    点击的位置。可选，默认 true 。

  - \*true\*: 点击本节点中刚刚识别到的目标（即点击自身）。

  - \*string\*: 填写节点名，点击之前执行过的某节点识别到的目标。

  - \*array<int, 4>\*: 点击固定坐标区域内随机一点，[x, y, w, h]，若希望全屏可设为 [0, 0, 0, 0] 。

- `target\_offset`: \*array<int, 4>\*

    在 `target` 的基础上额外移动再点击，四个值分别相加。可选，默认 [0, 0, 0, 0] 。

## ### `Swipe`

线性滑动。

该动作属性需额外部分字段：

- `begin`: \*true\* | \*string\* | \*array<int, 4>\*

    滑动起点。可选，默认 true 。值同上述 `Click`.`target` 。

- `begin\_offset`: \*array<int, 4>\*

    在 `begin` 的基础上额外移动再作为起点，四个值分别相加。可选，默认 [0, 0, 0, 0] 。

- `end`: \*true\* | \*string\* | \*array<int, 4>\*

    滑动终点。可选，默认 true 。值同上述 `Click`.`target` 。

- `end\_offset`: \*array<int, 4>\*

    在 `end` 的基础上额外移动再作为终点，四个值分别相加。可选，默认 [0, 0, 0, 0] 。

- `duration`: \*uint\*

    滑动持续时间，单位毫秒。可选，默认 200 。

## ### `MultiSwipe`

多指线性滑动。

该动作属性需额外部分字段：

- `swipes`: \*list<object,>\*

    多个滑动的数组。必选。

    数组元素顺序没有影响，只基于 `starting` 确定顺序。

  - `starting`: \*uint\*

    滑动起始时间，单位毫秒。可选，默认 0 。

    `MultiSwipe` 额外字段，该滑动会在本 action 中第 `starting` 毫秒才开始。

  - `begin`: \*true\* | \*string\* | \*array<int, 4>\*

    滑动起点。可选，默认 true 。值同上述 `Click`.`target` 。

  - `begin\_offset`: \*array<int, 4>\*

    在 `begin` 的基础上额外移动再作为起点，四个值分别相加。可选，默认 [0, 0, 0, 0] 。

  - `end`: \*true\* | \*string\* | \*array<int, 4>\*

    滑动终点。可选，默认 true 。值同上述 `Click`.`target` 。

  - `end\_offset`: \*array<int, 4>\*

    在 `end` 的基础上额外移动再作为终点，四个值分别相加。可选，默认 [0, 0, 0, 0] 。

  - `duration`: \*uint\*

    滑动持续时间，单位毫秒。可选，默认 200 。

举例：

```jsonc

{

    "A": {

        "action": "MultiSwipe",

        "swipes":

            {

                "begin": xxx,

                "end": xxx

            },

            {

                "starting": 500,

                "begin": xxx,

                "end": xxx

            }

        ]

    }

}

```

## ### `Key`

按键。

- `key`: \*int\* | \*list<int, >\*

    要按的键，仅支持对应控制器的虚拟按键码。

  - [Adb 控制器](https://developer.android.com/reference/android/view/KeyEvent)

  - [Win32 控制器](https://learn.microsoft.com/en-us/windows/win32/inputdev/virtual-key-codes)

## ### `InputText`

输入文本。

- `input\_text`: \*string\*

    要输入的文本，部分控制器仅支持 ascii 。

## ### `StartApp`

启动 App 。

该动作属性需额外部分字段：

- `package`: \*string\*

    启动入口。必选。

    需要填入 package name 或 activity ，例如 `com.hypergryph.arknights` 或 `com.hypergryph.arknights/com.u8.sdk.U8UnityContext` 。

## ### `StopApp`

关闭 App 。

该动作属性需额外部分字段：

- `package`: \*string\*

    要关闭的程序。必选。

    需要填入 package name ，例如 `com.hypergryph.arknights` 。

## ### `StopTask`

停止当前任务链（MaaTaskerPostTask 传入的单个任务链）。

## ### `Command`

执行命令。

该动作属性需额外部分字段：

- `exec`: \*string\*

    执行的程序路径。必选。

- `args`: \*list<string,>\*

    执行的参数。可选。

    支持部分运行期参数替换：

  - `{ENTRY}`: 任务入口名。

  - `{NODE}`: 当前节点名。

  - `{IMAGE}`: 截图保存到文件的路径。该文件在进程退出前删除，若要持久保存请自行复制。

  - `{BOX}`: 识别命中的目标，格式为 `[x, y, w, h]`。

  - `{RESOURCE\_DIR}`: 最后一次加载的资源文件夹路径。

  - `{LIBRARY\_DIR}`: MaaFW 库所在的文件夹路径。

- `detach`: \*bool\*

    分离子进程，即不等待子进程执行完成，直接继续执行后面的任务。可选，默认 false。

举例：

```jsonc

{

    "NodeA": {

        "action": "Command",

        "exec": "Python",

        "args": [

            "{RESOURCE\_DIR}/my\_script/test.py",

            "Haha",

            "{IMAGE}",

            "{NODE}",

            "{BOX}"

        ]

    },

    "NodeB": {

        "action": "Command",

        "exec": "{RESOURCE\_DIR}/my\_exec/my\_exec.exe"

    }

}

```

实际将会执行命令

```bash

# NodeA

Python C:/MaaXXX/resource/my\_script/test.py Haha C:/temp/123.png NodeA [0,0,0,0]

# NodeB

C:/MaaXXX/resource/my\_exec/my\_exec.exe

```

## ### `Custom`

执行通过 `MaaResourceRegisterCustomAction` 接口传入的动作句柄。

该动作属性需额外部分字段：

- `custom\_action`: \*string\*

    动作名，同注册接口传入的识别器名。同时会通过 `MaaCustomActionCallback`.`custom\_action\_name` 传出。必选。

- `custom\_action\_param`: \*any\*

    动作参数，任意类型，会通过 `MaaCustomActionCallback`.`custom\_action\_param` 传出。可选，默认空 json，即 `{}` 。

- `target`: \*true\* | \*string\* | \*array<int, 4>\*

    同 `Click`.`target`，会通过 `MaaCustomActionCallback`.`box` 传出。可选，默认 true 。

- `target\_offset`: \*array<int, 4>\*

    同 `Click`.`target\_offset` 。

## 等待画面静止

等待画面静止。需连续一定时间 画面 \*\*没有较大变化\*\* 才会退出动作。

字段值为 uint 或 object，举例：

```jsonc

{

    "A": {

        "pre\_wait\_freezes": 500,

    },

    "B": {

        "pre\_wait\_freezes": {

            // more properties ...

        },

    },

}

```

若值为 object，可设置部分额外字段：

- `time`: \*uint\*

    连续 `time` 毫秒 画面 \*\*没有较大变化\*\* 才会退出动作。可选，默认 1 。

- `target`: \*true\* | \*string\* | \*array<int, 4>\*

    等待的目标。可选，默认 true。值同上述 `Click`.`target` 。

- `target\_offset`: \*array<int, 4>\*

    在 `target` 的基础上额外移动再作为等待目标，四个值分别相加。可选，默认 [0, 0, 0, 0] 。

- `threshold`: \*double\*

    判断“没有较大变化”的模板匹配阈值。可选，默认 0.95 。

- `method`: \*int\*

    判断“没有较大变化”的模板匹配算法，即 cv::TemplateMatchModes。可选，默认 5 。

    同 `TemplateMatch`.`method` 。

- `rate\_limit`: \*uint\*

    识别速率限制，单位毫秒。可选，默认 1000 。

    每次识别最低消耗 `rate\_limit` 毫秒，不足的时间将会 sleep 等待。

- `timeout`: \*uint\*

    识别超时时间，毫秒。可选，默认 20 \* 1000 。