

---

# Vehicles API docs

*Release 1.0-RC*

Sep 10, 2019



<b>1</b>	<b>Car noCar Classifier</b>	<b>1</b>
1.1	Description . . . . .	1
1.2	Example Request . . . . .	2
1.3	Example Response . . . . .	3
1.4	Authentication . . . . .	3
1.5	Performances and Limits . . . . .	4
<b>2</b>	<b>Vehicle Type Classifier</b>	<b>5</b>
2.1	Description . . . . .	5
2.2	Example Request . . . . .	6
2.3	Example Response . . . . .	7
2.4	Authentication . . . . .	7
2.5	Performances and Limits . . . . .	8



### 1.1 Description

This API provides an image classifier which understands if an image **contains a car or not**.

#### 1.1.1 Request:

The API expects a POST request providing an image, encoded in base64 format.

Learn [here](#) about the base64 format.

Find [here](#) an online converter.

POST to: <https://api.cloudif.ai/cnc/predict>

#### 1.1.2 Response

The API returns an array of JSON Object with the label and probability score associated in the following order:

- car
- no\_car
- other

Here's the OpenAPI 3 specification.

```
openapi: 3.0.0
info:
  title: Vehicle Type
  description: Vehicle Type Classification API.
  version: 0.1.9
servers:
  - url: https://api.cloudif.ai/cnc/predict
```

(continues on next page)

(continued from previous page)

```

description: Production server

paths:
  /predict-vehicle:
    post:
      summary: Requires a base64 encoded image as body request, returns the predicted_
↪vehicle type.
      description:

        It expect as input a base64 encoded image, black & white images are not_
↪allowed.
        To encode the image you can use a webtool like https://www.base64decode.org/

        Returns a list of float with the prediction confidence for the given class._
↪The available classes have this positional order [car, no-car].

      requestBody:
        description: Encode the vehicle image you want to classify.
        required: true
        content:
          application/json:
            schema:
              type: object
              properties:
                base64:
                  type: string
                  format: int64
                  example:
↪"iVBORw0KGgoAAAANSUhEUgAAAAoAAAAKCAAAAACNMs+9AAAFU1EQVR42mNc85+hnoEIwDlqkL4KAezxFa8zNQrMAAAAAE1FTI
↪"

        responses:
          '200': # status code
            description: A JSON array of class probabilities
            content:
              application/json:
                schema:
                  type: array
                  items:
                    type: string

          '400': # status code
            description: 'Bad request.'

```

## 1.2 Example Request

Here's an example request made with Python 3. Remember to replace `YOUR_BASE64_STRING` and `YOUR_API_KEY` with current values Remember to change your API Key in the following code.

```

import http.client

conn = http.client.HTTPConnection("api.cloudif.ai")

payload = "{\n  \"base64\": \"YOUR_BASE64_STRING\"

```

(continues on next page)

(continued from previous page)

```
headers = {
  'Content-Type': "application/json",
  'x-api-key': "YOUR_API_KEY",
  'User-Agent': "PostmanRuntime/7.16.3",
  'Accept': "*/*",
  'Cache-Control': "no-cache",
  'Postman-Token': "8f05ff29-7a87-44ba-a88c-6a11162db39e,55d71da6-56ed-43ca-bb24-
↪0589a11dfb91",
  'Host': "api.cloudif.ai",
  'Accept-Encoding': "gzip, deflate",
  'Content-Length': "8114",
  'Connection': "keep-alive",
  'cache-control': "no-cache"
}

conn.request("POST", "cnc.predict", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

## 1.3 Example Response

Here's an example response.

```
{
  "response": {
    "status": "success",
    "prediction": "car",
    "results": [
      {
        "score": 0.974482,
        "label": "car"
      },
      {
        "score": 0.0255056,
        "label": "no-car"
      }
    ]
  }
}
```

## 1.4 Authentication

Use your private API key.

## 1.5 Performances and Limits

Limit size for upload: 2 MegaByte



---

## Vehicle Type Classifier

---

### 2.1 Description

This API provides an image classifier which understands if an image **is a car, a motorcycle or a business vehicle**.

#### 2.1.1 Request:

The API expects a POST request providing an image, encoded in base64 format.

Learn [here](#) about the base64 format.

Find [here](#) an online converter.

POST to:

#### 2.1.2 Response

The API returns an array of JSON Object with the label and probability score associated in the following order:

- car
- motorcycle
- business vehicle

Here's the OpenAPI 3 specification.

```
openapi: 3.0.0
info:
  title: Vehicle Type
  description: Vehicle Type Classification API.
  version: 0.1.9
servers:
```

(continues on next page)

(continued from previous page)

```

- url: https://api.cloudif.ai/vehicle-type/predict
  description: Production server

paths:
  /predict-vehicle:
    post:
      summary: Requires a base64 encoded image as body request, returns the predicted
      ↪vehicle type.
      description:

        It expect as input a base64 encoded image, black & white images are not
      ↪allowed.
        To encode the image you can use a webtool like https://www.base64decode.org/

        Returns a list of float with the prediction confidence for the given class.
      ↪The available classes have this positional order [car, motorbike, business vehicle].

      requestBody:
        description: Encode the vehicle image you want to classify.
        required: true
        content:
          application/json:
            schema:
              type: object
              properties:
                base64:
                  type: string
                  format: int64
                  example:
      ↪"iVBORw0KGgoAAAANSUgAAAAoAAAKCAYAAACNmS+9AAAFU1EQVR42mNc85+hnoEIwDlqkL4KAezxFa8zNQrMAAAAAE1FT
      ↪"

      responses:
        '200': # status code
          description: A JSON array of class probabilities
          content:
            application/json:
              schema:
                type: array
                items:
                  type: string

        '400': # status code
          description: 'Bad request.'

```

## 2.2 Example Request

Here's an example request made with Python 3. Remember to replace `YOUR_BASE64_STRING` and `YOUR_API_KEY` with current values Remember to change your API Key in the following code.

```

import http.client

conn = http.client.HTTPConnection("api.cloudif.ai")

```

(continues on next page)

(continued from previous page)

```
payload = "{\n  \"base64\": \"YOUR_BASE64_STRING\"

headers = {
  'Content-Type': \"application/json\",
  'x-api-key': \"YOUR_API_KEY\",
  'User-Agent': \"PostmanRuntime/7.16.3\",
  'Accept': \"*/*\",
  'Cache-Control': \"no-cache\",
  'Postman-Token': \"8f05ff29-7a87-44ba-a88c-6a11162db39e,55d71da6-56ed-43ca-bb24-
↪0589a11dfb91\",
  'Host': \"api.cloudif.ai\",
  'Accept-Encoding': \"gzip, deflate\",
  'Content-Length': \"8114\",
  'Connection': \"keep-alive\",
  'cache-control': \"no-cache\"
}

conn.request(\"POST\", \"vehicle-type.predict\", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode(\"utf-8\"))
```

## 2.3 Example Response

Here's an example response.

```
{
  "response": {
    "status": "success",
    "prediction": "car",
    "results": [
      {
        "score": 0.974482,
        "label": "car"
      },
      {
        "score": 0.0255056,
        "label": "motorcycle"
      },
      {
        "score": 1.24175e-05,
        "label": "business"
      }
    ]
  }
}
```

## 2.4 Authentication

Use your private API key.

## 2.5 Performances and Limits

Limit size for upload: 2 MegaByte