Object Detection

Run inference on your object detection models hosted on Roboflow.

There are several ways to run object detection inferences using the Roboflow Hosted API. You can use one of our different SDKs, or send a REST request to our hosted endpoint.

Python | cU... | Javascri... | Swift/i... | Andr... | Ru... | P... | G... | .N... | Mo...

To run inference through our hosted API using Python, use the `roboflow` Python package:

```
from roboflow import Roboflow
rf = Roboflow(api_key="API_KEY")
project = rf.workspace().project("MODEL_ENDPOINT")
model = project.version(VERSION).model

# infer on a local image
print(model.predict("your_image.jpg", confidence=40, overlap=30).json())

# visualize your prediction
# model.predict("your_image.jpg", confidence=40, overlap=30).save("predictio

# infer on an image hosted elsewhere
# print(model.predict("URL_OF_YOUR_IMAGE", hosted=True, confidence=40, overla
```

Using the Inference API

`POST` `https://detect.roboflow.com/:datasetSlug/:versionNumber`

Path Parameters

Name	Type	Description
datasetSlug	string	The url-safe version of the dataset name. You can f the web UI by looking at the URL on the main projec

Ask AI

Name	Type	Description
		by clicking the "Get curl command" button in the train results section of your dataset version after training your model.
version	number	The version number identifying the version of of your dataset

Query Parameters

Name	Type	Description
image	string	URL of the image to add. Use if your image is hosted elsewhere. (Required when you don't POST a base64 encoded image in the request body.) **Note:** don't forget to URL-encode it.
classes	string	Restrict the predictions to only those of certain classes. Provide as a comma-separated string. **Example:** dog,cat **Default:** not present (show all classes)
overlap	number	The maximum percentage (on a scale of 0-100) that bounding box predictions of the same class are allowed to overlap before being combined into a single box. **Default:** 30
confidence	number	A threshold for the returned predictions on a scale of 0-100. A lower number will return more predictions. A higher number will return fewer high-certainty predictions. **Default:** 40
stroke	number	The width (in pixels) of the bounding box displayed around predictions (only has an effect when `format` is `image`). **Default:** 1
labels	boolean	Whether or not to display text labels on the predictions (only has an effect when `format` is `image`).

Name	Type	Description
		Default: false
format	string	**json** - returns an array of JSON predictions. (See response format tab). **image** - returns an image with annotated predictions as a binary blob with a `Content-Type` of `image/jpeg`. **image_and_json** - returns an array of JSON predictions, including a visualization field in base64. **Default**: json
api_key	string	Your API key (obtained via your workspace API settings page)

Request Body

Name	Type	Description
	string	A base64 encoded image. (Required when you don't pass an image URL in the query parameters).

200 JSON format predictions. (x,y) are the box's center pixel coordinates. 403 If your api_key...

```
{
    "predictions": [{
        "x": 234.0,
        "y": 363.5,
        "width": 160,
        "height": 197,
        "class": "hand",
        "confidence": 0.943
    }, {
        "x": 504.5,
        "y": 363.0,
        "width": 215,
        "height": 172,
        "class": "hand",
        "confidence": 0.917
    }, {
        "x": 1112.5,
        "y": 691.0,
        "width": 139,
        "height": 52,
        "class": "hand",
        "confidence": 0.87
    }, {
        "x": 78.5,
        "y": 700.0,
        "width": 139,
        "height": 34,
        "class": "hand",
        "confidence": 0.404
    }]
}
```


API Reference

URL

`POST` `https://detect.roboflow.com/:projectId/:versionNumber`

Name	Type	Description
projectId	string	jO8taetGlQqV

Name	Type	Description
version	number	eM90DCAbt4yj

See how to get your project ID and version number here.

There are two ways you can send an image to the Hosted Inference API via a REST request:

- Attach a `base64` encoded image to the `POST` request body
- Send a URL of an image file using the `image` URL query
  - ex:
    ```
    https://detect.roboflow.com/:datasetSlug/:versionNumber?
    image=https://imageurl.com
    ```

Query Parameters

Name	Type	Description
image	string	ywU24xjaL8Xj
classes	string	aewuq4lFpSmc
overlap	number	lCgdmy9Uqaki
confidence	number	zaavsEGYKhq9
stroke	number	OzSpHhFy2xmS
labels	boolean	tX7ZHSWeOPdN
format	string	**Options:** • **json:** returns an array of JSON predictions. (See response format tab). • **image:** returns an image with annotated predictions as a binary blob with a `Content-Type` of `image/jpeg`. **Default**: `json`
api_key	string	lSdGUlSM2n8W

Body

Type	Description
	JaAkhBEp585w

The content type should be `application/x-www-form-urlencoded` with a string body.

Response Format

The hosted API inference endpoint, as well as most of our SDKs, return a `JSON` object containing an array of predictions. Each prediction has the following properties:

- `x` = the horizontal center point of the detected object
- `y` = the vertical center point of the detected object
- `width` = the width of the bounding box
- `height` = the height of the bounding box
- `class` = the class label of the detected object
- `confidence` = the model's confidence that the detected object has the correct label and position coordinates

Here is an example response object from the REST API:

```
{
    "predictions": [
        {
            "x": 189.5,
            "y": 100,
            "width": 163,
            "height": 186,
            "class": "helmet",
            "confidence": 0.544
        }
    ],
    "image": {
        "width": 2048,
        "height": 1371
    }
}
```

The `image` attribute contains the height and width of the image sent for inference. You may need to use these values for bounding box calculations.

Drawing a Box from the Inference API JSON Output

Frameworks and packages for rendering bounding boxes can differ in positional formats. Given the response `JSON` object's properties, a bounding box can always be drawn using some combination of the following rules:

- the center point will always be (`x` , `y`)
- the corner points `(x1, y1)` and `(x2, y2)` can be found using:
  - `x1` = `x - (width/2)`
  - `y1` = `y - (height/2)`
  - `x2` = `x + (width/2)`
  - `y2` = `y + (height/2)`

The corner points approach is a common pattern and seen in libraries such as `Pillow` when building the `box` object to render bounding boxes within an `Image`.

Don't forget to iterate through all detections found when working with `predictions`!

```
# example box object from the Pillow library
for bounding_box in detections:
    x1 = bounding_box['x'] - bounding_box['width'] / 2
    x2 = bounding_box['x'] + bounding_box['width'] / 2
    y1 = bounding_box['y'] - bounding_box['height'] / 2
    y2 = bounding_box['y'] + bounding_box['height'] / 2
    box = (x1, x2, y1, y2)
```

Last updated 1 month ago