

COMPUTER VISION EDUCATION OBJECT DETECTION

How to Draw a Bounding Box for Computer Vision with Python

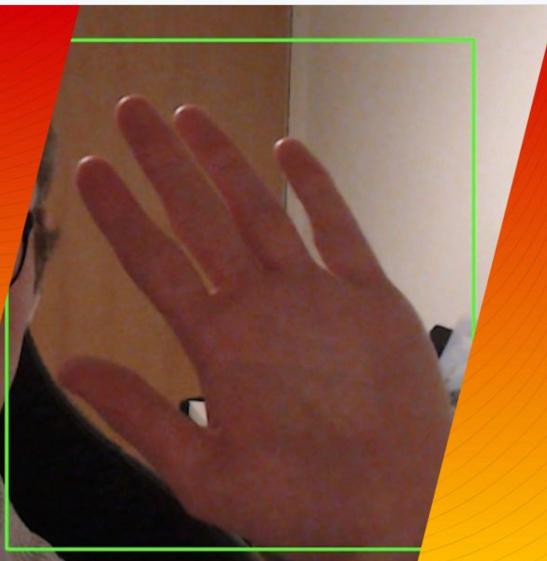


James Gallagher

JAN 19, 2023 | 5 MIN READ

How to Draw a Bounding Box for Computer Vision with Python

roboflow

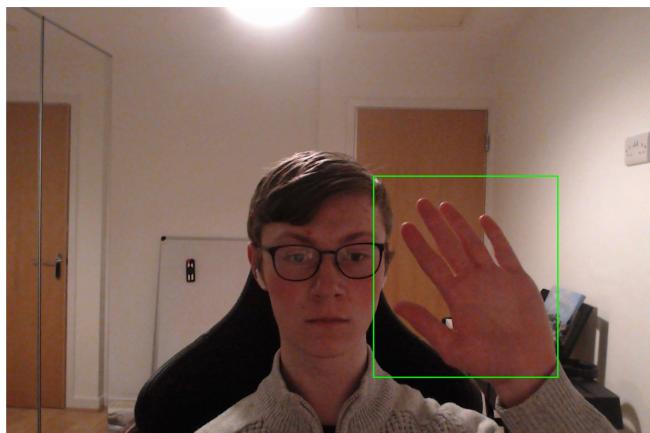


When you see computer vision models in demos, you'll usually see boxes drawn around predictions. These are called bounding boxes. Bounding boxes help visualize the results of a model.

Using bounding boxes, you can evaluate how your model performs (i.e. if there are certain predictions it misses, or certain objects that tend to come up as false positives more). In combination with understanding class balance in your dataset, what you learn from testing your model and analyzing the bounding boxes will be helpful to improving model performance.

But, bounding boxes are not drawn on photos by the model itself. This is a post-processing step that may or may not be done by a model. When you use the Roboflow Python package, for example, you may opt only to parse the JSON output of a model and opt to draw your own bounding boxes.

Here's an example image with a bounding box that we'll create in this tutorial:



Build and deploy with Roboflow for free

Use Roboflow to manage datasets, train models in one-click, and deploy to web, mobile, or the edge.

[Try It Now](#)

SUBSCRIBE TO OUR NEWSLETTER

 Enter email[Subscribe](#)

Unsubscribe at any time. Review our [Privacy Policy](#).

TABLE OF CONTENTS

How to Draw a Bounding Box in Python

[How to Fill a Bounding Box](#)[Setting Up a Model \(Optional\)](#)[Conclusion](#)

RECOMMENDED READS

[Distilling GPT-4 for Classification with an API](#)

[DINO-GPT4-V: Use GPT-4V in a Two-Stage Detection Model](#)

[How CLIP and GPT-4V Compare for Classification](#)

In this article, we're going to talk about how to draw a bounding box on an image in Python. Without further ado, let's begin!

How to Draw a Bounding Box in Python

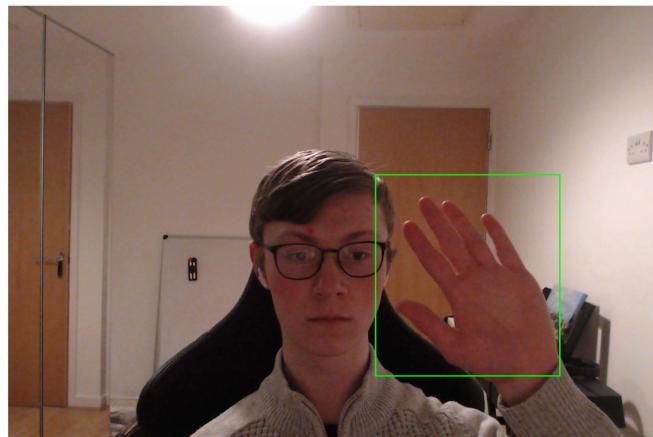
To draw a bounding box in Python, we need four coordinates: one coordinate representing each corner of a bounding box. The Roboflow API, for example, provides an x and y coordinate alongside the height and width of a bounding box.

With this information, we can calculate the coordinates for each corner of the box and visualize a bounding box. Here is the code we need:

```
1 import cv2
2
3 predictions = {'predictions': [{'x': 1012.0, 'y': 593.5, 'width': 406.0, 'height': 443.0,
4
5     for bounding_box in predictions["predictions"]:
6         x0 = bounding_box['x'] - bounding_box['width'] / 2
7         x1 = bounding_box['x'] + bounding_box['width'] / 2
8         y0 = bounding_box['y'] - bounding_box['height'] / 2
9         y1 = bounding_box['y'] + bounding_box['height'] / 2
10
11         start_point = (int(x0), int(y0))
12         end_point = (int(x1), int(y1))
13         cv2.rectangle(img, start_point, end_point, color=(0,255,0), thickness=2)
14
15 cv2.imwrite("example_with_bounding_boxes.jpg", image)
```

You will need to add in your own list of predictions with the x, y, width and height values that represent the bounding box that you want to draw. For this example, we have added a prediction returned by the Roboflow model we discuss at the end of this article.

Let's run our code! After doing so, we can open up our "example_with_bounding_boxes.jpg" file to see the result of our prediction. Here are the results:



We successfully have a green bounding box on our image!

The color=() tuple accepts RGB colors. We can thus change the color of the borders of our bounding boxes by changing the value of the tuple. For instance, if we want to show blue bounding boxes, we can use this code:

```
cv2.rectangle(image, start_point, end_point, color=(0,0,255), thickness=2)
```

How to Fill a Bounding Box

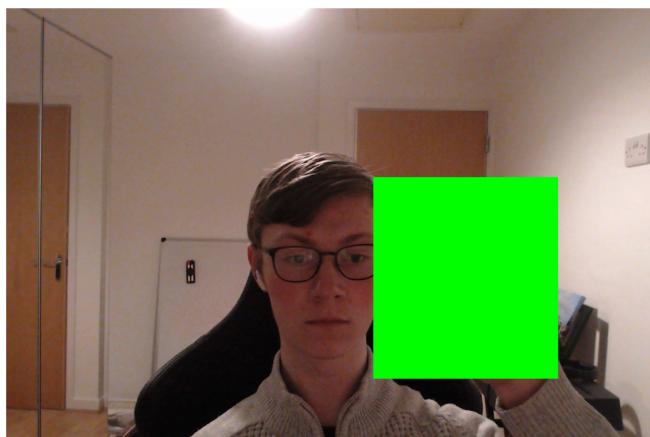
You may want to fill a bounding box with color in situations where the contents of a bounding box are sensitive. For example, if you want to hide license plates in an image, you may want to fill the bounding box with a color so that they are not visible.

To fill a bounding box, use this line of code:

```
cv2.rectangle(image, start_point, end_point, color=(0,255,0), thickness=-1)
```

The thickness=-1 property tells cv2 to create and show a bounding box full of the specified color. In the above code, we specify that we want our bounding box to be green. As a result, the contents of the box will be green.

Let's run our code and open our "example_with_bounding_boxes.jpg" image again to see the results:



As you can see, the prediction in our image is hidden by the green box.

Setting Up a Model (Optional)

For this tutorial, we're going to use the [Rock Paper Scissors](#) project on Roboflow Universe, a repository of over 110,000 public computer vision datasets. This project has a hosted API already available for us, so we don't need to do any model preparation work. All we need is a Roboflow account, the API key from which we will use to import our model into a Python script.

If you have a model already working in your code, you can skip this section. There is no requirement to use Roboflow for our bounding box guide.

First, let's install the Roboflow Python package:

```
pip install roboflow
```

Then, let's import the Rock Paper Scissors model into our code. We can do this by copying the code snippet that Roboflow Universe provides on the "Deploy" tab, accessible from the sidebar of the model page. The deploy tab will give us some Python code that looks like this:

```
1 from roboflow import Roboflow
2
3 rf = Roboflow(api_key="KEY")
4 project = rf.workspace().project("rock-paper-scissors-detection")
5 model = project.version(34).model
```

You will need to substitute "KEY" for your Roboflow API key, which will be in the snippet you copy from Roboflow Universe. Alternatively, you can follow our guide on how to retrieve your API key. Now we have a model ready for use!

To retrieve predictions on an image, we can use this code:

```
1 predictions = model.predict("image1.jpg", confidence=70).json()
2 print(predictions)
```

This code will create a JSON object that contains all predictions made by our model on "image1.jpg". Only predictions with a confidence level greater than 70% will be displayed. You can edit the confidence level to suit your needs.

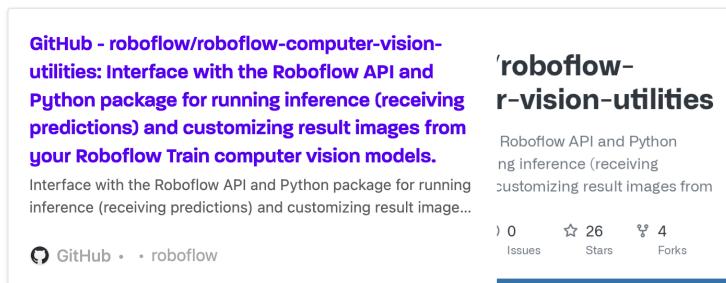
Here is an example prediction:

```
{'predictions': [{}x': 1012.0, 'y': 593.5, 'width': 406.0, 'height': 443.0, 'confidence':
```

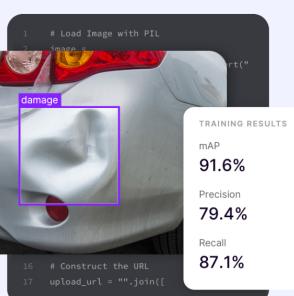
This object shows that one prediction with the Paper class was returned by the model. We can use this dictionary of predictions with our code above to draw bounding boxes on our image.

Conclusion

In this guide, we have discussed how to draw a bounding box in Python using the cv2 library. This will enable you to visualize the predictions returned from your computer vision model. You can customize the color of bounding box borders. You can also fill the contents of a bounding box to hide content, such as may be required to redact sensitive information from an image.



Now you have the tools you need to successfully draw and fill a bounding box in Python!



Build and deploy computer vision models with Roboflow

Join over 100,000 developers and top-tier companies from Walmart to Cardinal Health building computer vision models with Roboflow.

[Get started](#)



James Gallagher

James is a Technical Marketer at Roboflow, working toward democratizing access to computer vision.

[VIEW MORE POSTS](#)

TOPICS:

Computer Vision, Education, Object Detection

MORE ABOUT COMPUTER VISION[VIEW ALL](#)**Distilling GPT-4 for Classification with an API**

NOV 7, 2023

**DINO-GPT4-V: Use GPT-4V in a Two-Stage Detection Model**

NOV 7, 2023

**How CLIP and GPT-4V Compare for Classification**

NOV 7, 2023

**Experiments with GPT-4V for Object Detection**

NOV 7, 2023

**How to Use MetaCLIP**

NOV 3, 2023

**How to Provide Detailed Labeling Instructions to Outsourced Labelers**

NOV 3, 2023

Want to learn more about Roboflow? Email sales@roboflow.com or [book a demo](#) with our sales team.

© 2023 Roboflow, Inc.
All rights reserved.

For sales inquiries:

[✉️ sales@roboflow.com](mailto:sales@roboflow.com)
[Book a demo](#)

**PRODUCT**[Sign In / Sign Up](#)[Universe](#)[Annotate](#)[Train](#)[Deploy](#)[Integrations](#)[Pricing](#)**ECOSYSTEM**[Notebooks](#)[Autodistill](#)[Supervision](#)[Roboflow 100](#)**DEVELOPERS**[User Forum](#)[Templates](#)[Blog](#)[Contact Sales](#)[Learn Computer Vision](#)[Convert Annotation Formats](#)[Computer Vision Models](#)[Computer Vision Utilities](#)**INDUSTRIES**[Manufacturing](#)[Oil & Gas](#)[Retail](#)[Safety & Security](#)[Transportation](#)[All Industries](#)**COMPANY**[About Us](#)[Careers](#)[Press](#)[Media Kit](#)[Terms of Service](#)[Privacy Policy](#)[Sitemap](#)