

COMPUTER VISION EDUCATION

How to Draw a Bounding Box Prediction Label with Python



James Gallagher

JAN 19, 2023 | 3 MIN READ

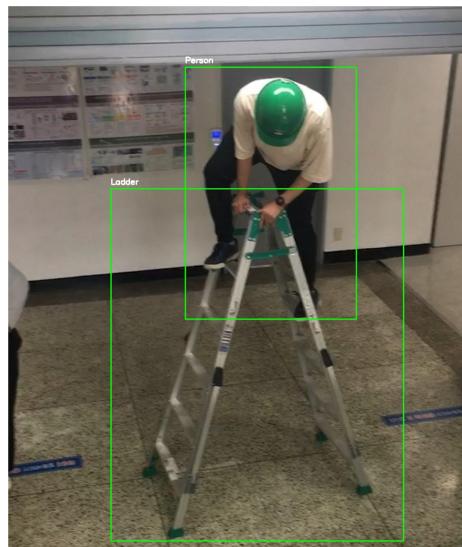
How to Draw a Bounding Box Prediction Label with Python



Computer vision models, by default, return the positions of predictions returned by the model, not a visual bounding box on an image. Drawing boxes showing the location of predictions, and adding labels to those boxes, is a post-processing step which requires additional code.

Once you have drawn bounding boxes – a topic about which we have written in the past – you may be wondering: how can I visualize a prediction? That's the question we're going to answer in this tutorial.

Let's discuss how to draw a bounding box prediction using Python and the cv2.putText method. Here is the end result we'll achieve:



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 Label in Python

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](#)

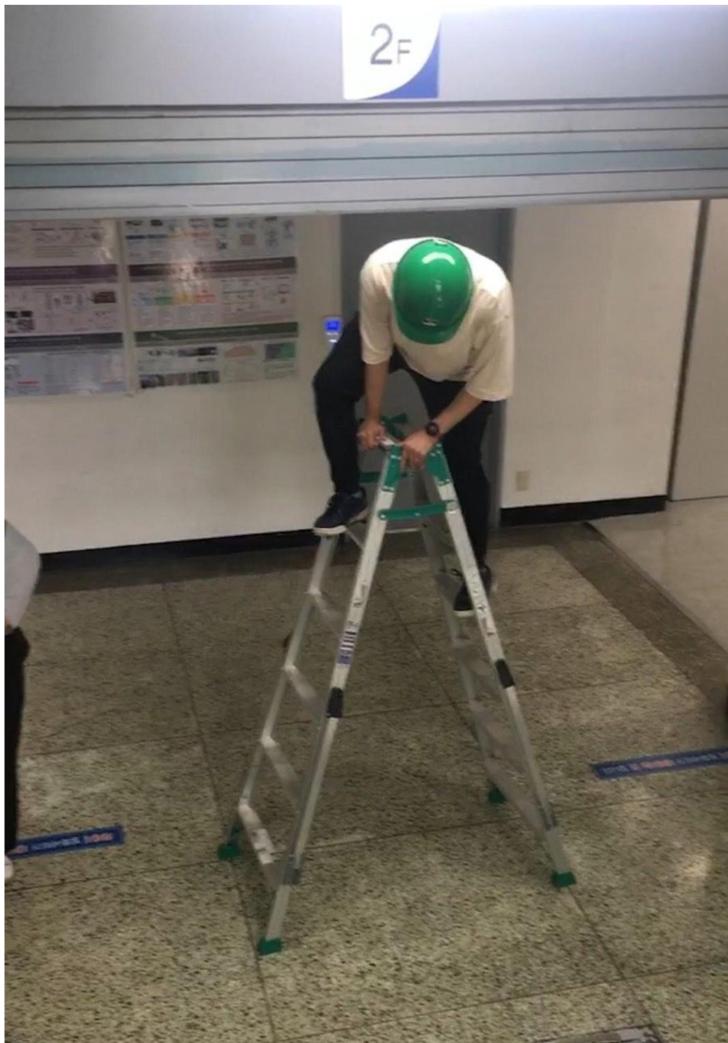
How to Draw a Bounding Box Label in Python

If you don't already have a model, scroll down to the bottom of this article where we'll show using an example model from Roboflow Universe.

To draw a bounding box, you'll need:

1. The starting x position of the box.
2. The starting y position of the box.
3. The width of the prediction.
4. The height of the prediction.

Here is the raw image to which we want to add bounding boxes and labels:



Below, we use the "predictions" list from the last section which contains all of this information. You can substitute the variables below as necessary to make the code work on your project. Here is the code we'll need to draw our bounding boxes and add labels above each bounding box:

```
1 import cv2
2
3 predictions = [...] # add your predictions here
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=(0,0,0), thickness=1)
14
15    cv2.putText(
```

```

16     image,
17     bounding_box["class"],
18     (int(x0), int(y0) - 10),
19     fontFace = cv2.FONT_HERSHEY_SIMPLEX,
20     fontScale = 0.6,
21     color = (255, 255, 255),
22     thickness=2
23   )
24
25 cv2.imwrite("example_with_bounding_boxes.jpg", image)

```

This code will add a label with the predicted class name above our image. The (x, y) tuple represents the place in the image where we want to add our label. We then specify a few other attributes: the typeface to use, the font scale, the color of the text (in our case above, black), and the weight of the text (how bold it is).

Let's run our code using the ladders detection model (discussed in the next section):



Now all of our bounding boxes show a label at the top.

Setting Up a Model (Optional)

In the guide above, we drew bounding box labels around a model that detects people and ladders. If you want to use the people and ladder detection models for testing, first sign up to Roboflow and then go to our [People and Ladders Universe dataset](#). This dataset comes with a free-to-use trained model.

Click "Model" in the sidebar, scroll down and copy the code that you see into a Python file. The code will look something like this, with your API key:

```

1  from roboflow import Roboflow
2
3  rf = Roboflow(api_key="API KEY")

```

```
4 project = rf.workspace().project("people-and-ladders")
5 model = project.version(4).model
```

This code imports your model. Add a new line at the bottom to run inference on a local file:

```
predictions = model.predict("image.jpg", confidence=50).json()
```

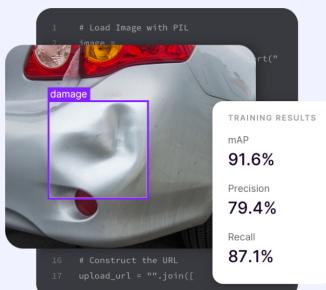
The "predictions" variable now contains all of the predictions returned by our model that were rated with a confidence level equal to or greater than 50%. Now we have a model on whose predictions we can draw bounding boxes and labels.

Conclusion

In this tutorial, we have discussed:

1. How to use Roboflow Universe to use a trained public model;
2. How to retrieve predictions from the model;
3. How to draw bounding boxes around a model and;
4. How to draw labels above each bounding box.

Now you have all of the information you need to represent numerical coordinates data visually in your computer vision work. Happy creating!



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

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](#)

