

Detect objects in images

1. Introduction

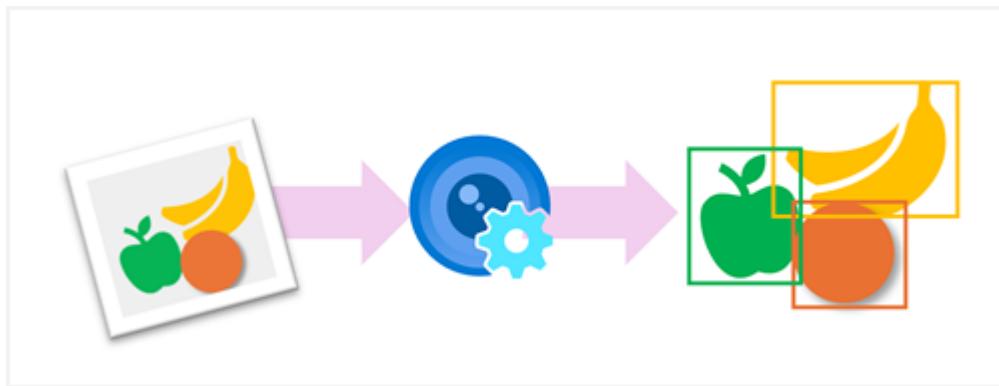
<https://learn.microsoft.com/en-us/training/modules/detect-objects-images/1-introduction>

Introduction

Completed

- 1 minute

Object detection is a common computer vision problem that requires software to identify the location of specific classes of object in an image.



For example, an automated checkout system in a grocery store might use a camera to monitor a checkout conveyer belt on which there might be multiple different items at any one time. The system could use object detection to identify which items are on the belt, and where in the image they appear.

In this module, you'll learn how to use the Azure AI Custom Vision service to create object detection models.

2. Use Azure AI Custom Vision for object detection

Use Azure AI Custom Vision for object detection

Completed

- 5 minutes

To use the Custom Vision service to create an object detection solution, you need *two* Custom Vision resources in your Azure subscription:

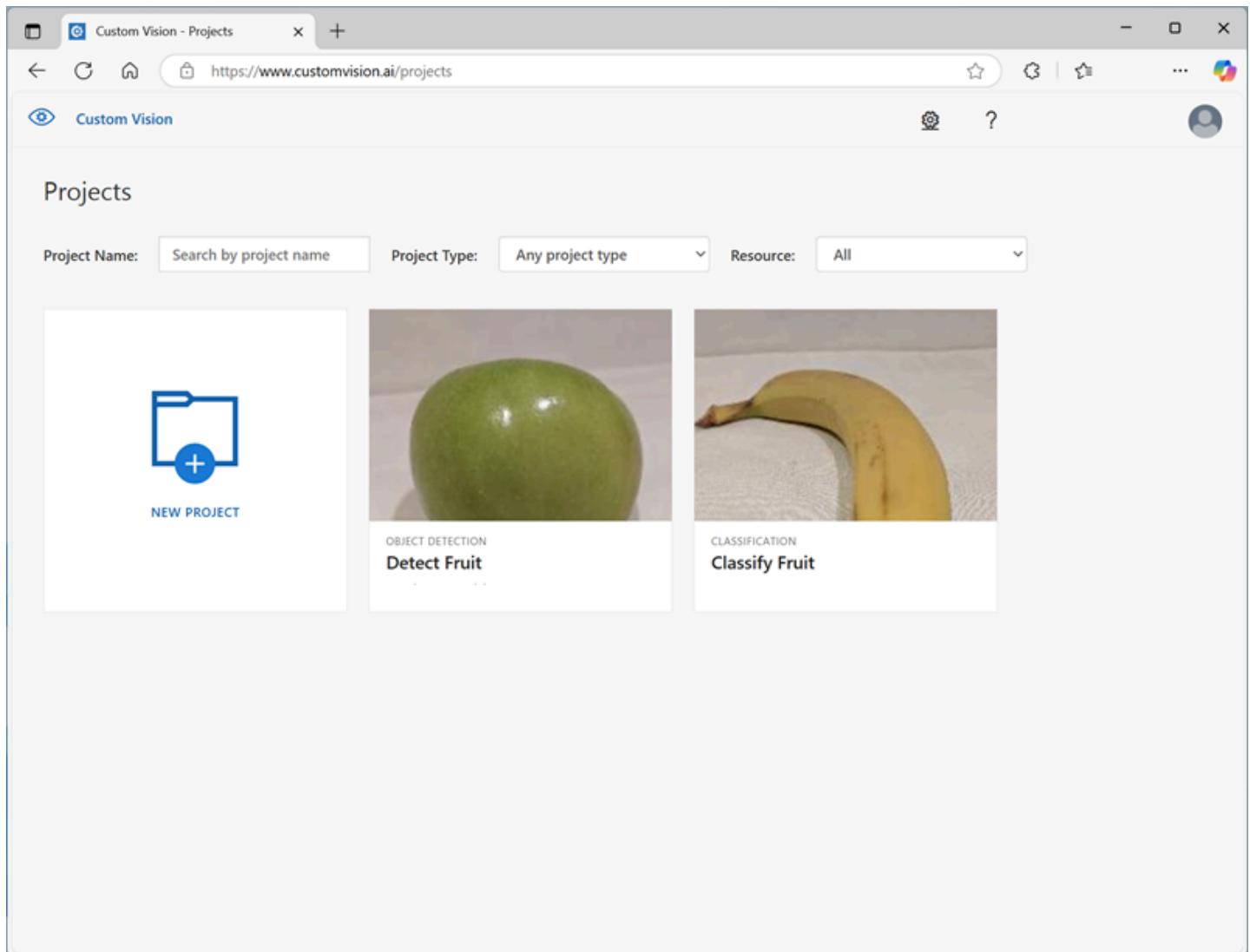
- An **Azure AI Custom Vision *training*** resource - used to train a custom model based on your own training images.
- An **Azure AI Custom Vision *prediction*** resource - used to generate predictions from new images based on your trained model.

When you provision the Azure AI Custom Vision service in an Azure subscription, you can choose to create one or both of these resources. This separation of training and prediction provides flexibility. For example, you can use a *training* resource in one region to train your model using your own image data; and then deploy one or more prediction resources in other regions to support computer vision applications that need to use your model.

Each resource has its own unique *endpoint* and authentication *keys*; which are used by client applications to connect and authenticate to the service.

The Custom Vision portal

Azure AI Custom Vision provides a web-based portal, in which you can train, publish, and test custom vision models.



You can sign into the Custom Vision portal at <https://www.customvision.ai/> using your Azure credentials and use it to create image classification or object detection projects that use Azure AI Custom Vision resources in your Azure subscription.

Each project has a unique *project ID*; which is used by client applications to perform training or prediction tasks using code.

Custom Vision SDKs

You can write code to train and consume custom models by using the Azure AI Custom Vision language-specific SDKs.

For example, Microsoft C# developers can use the [Microsoft.Azure.CognitiveServices.Vision.CustomVision.Training](#) and [Microsoft.Azure.CognitiveServices.Vision.CustomVision.Prediction](#) Microsoft .NET packages for training and prediction respectively.

Python developers can perform both training and prediction tasks by using the [azure-cognitiveservices-vision-customvision](#) package.

3. Train an object detector

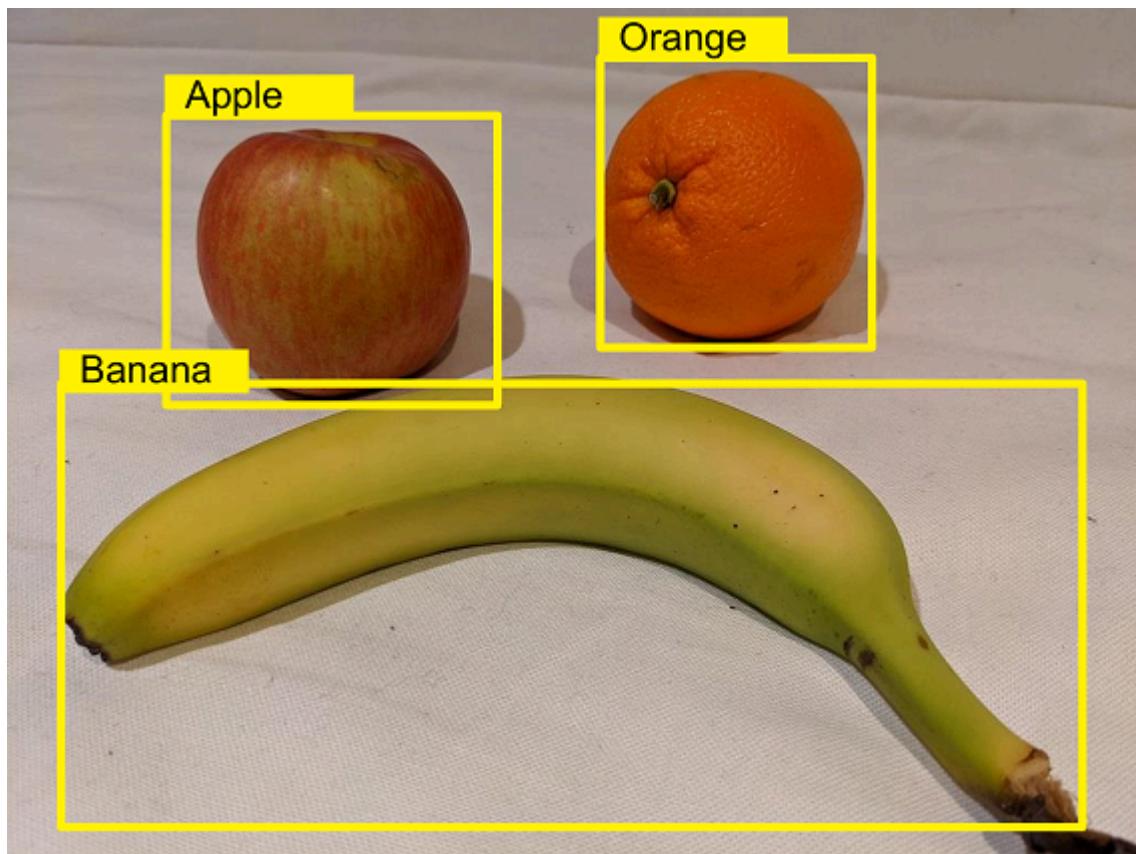
<https://learn.microsoft.com/en-us/training/modules/detect-objects-images/3-train-object-detector>

Train an object detector

Completed

- 5 minutes

Object detection is a form of computer vision in which a model is trained to detect the presence and location of one or more classes of object in an image.



There are two components to an object detection prediction:

- The class label of each object detected in the image. For example, you might ascertain that an image contains an apple, an orange, and a banana.
- The location of each object within the image, indicated as coordinates of a *bounding box* that encloses the object.

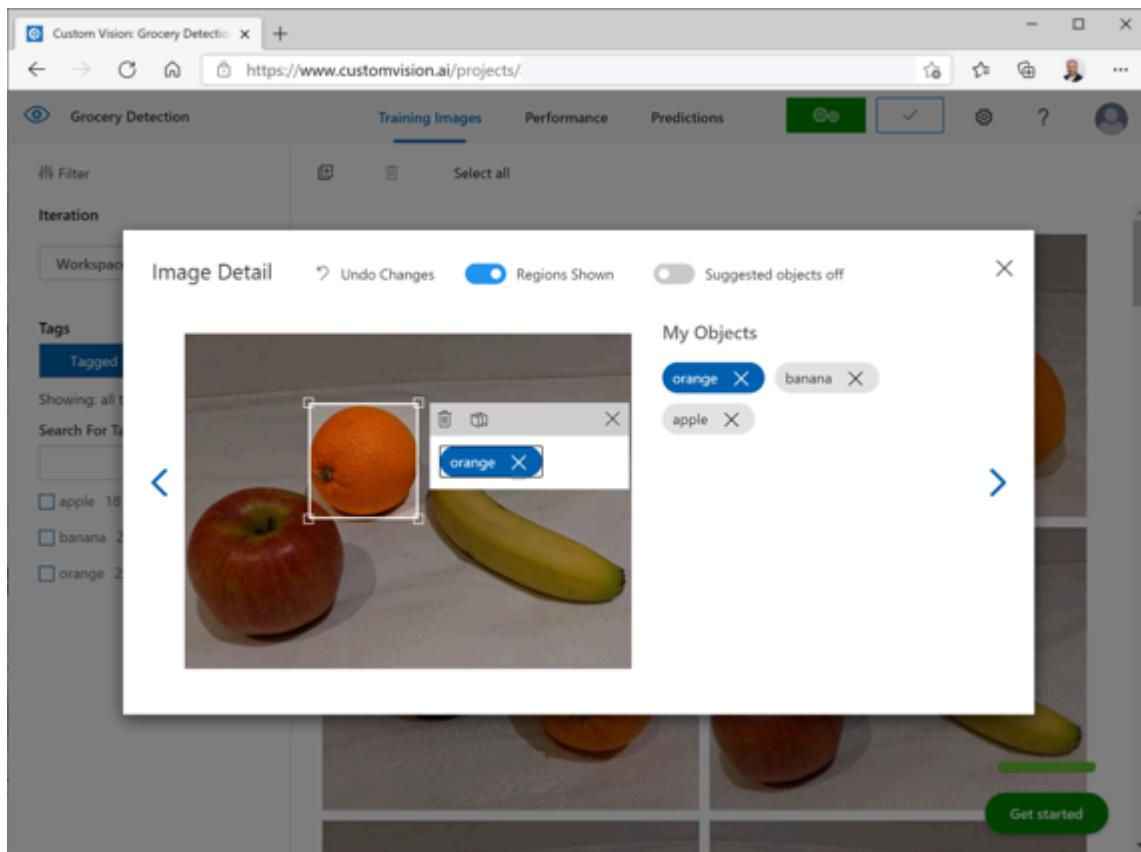
To train an object detection model, you can use the Azure AI Custom Vision portal to upload and label images before training, evaluating, testing, and publishing the model; or you can use the REST API or a language-specific SDK to write code that performs the training tasks.

Image labeling

You can use Azure AI Custom Vision to create projects for *image classification* or *object detection*. The most significant difference between training an image classification model and training an object detection model is the labeling of the images with tags. While image classification requires one or more tags that apply to the whole image, object detection requires that each label consists of a tag and a *region* that defines the bounding box for each object in an image.

Labeling images in the Azure AI Custom Vision portal

The Azure AI Custom Vision portal provides a graphical interface that you can use to label your training images.



The easiest option for labeling images for object detection is to use the interactive interface in the Azure AI Custom Vision portal. This interface automatically suggests regions that contain objects, to which you can assign tags or adjust by dragging the bounding box to enclose the object you want to label.

Additionally, after tagging an initial batch of images, you can train the model. Subsequent labeling of new images can benefit from the *smart labeler* tool in the portal, which can suggest not only the

regions, but the classes of object they contain.

Alternative labeling approaches

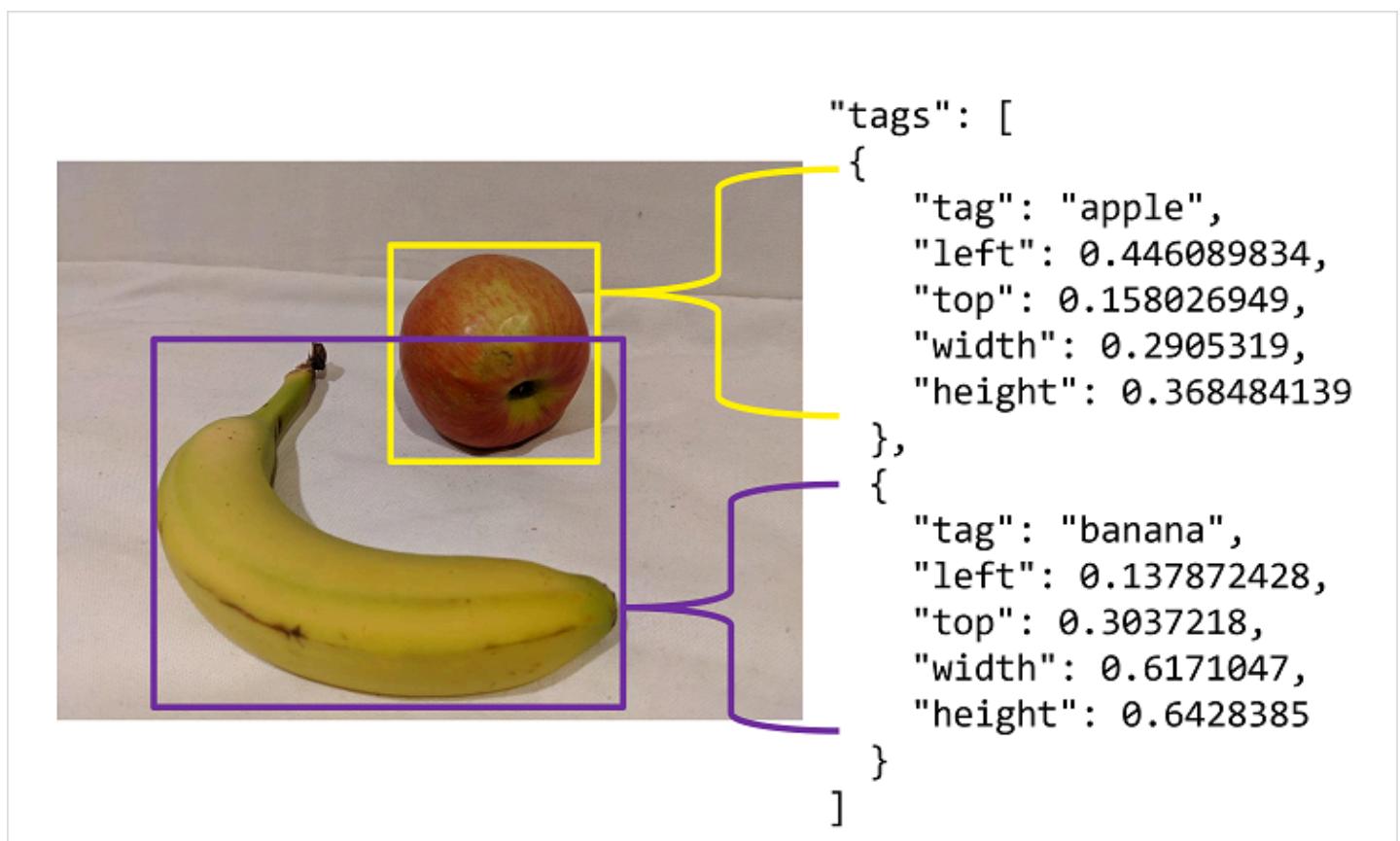
Alternatively, you can use a custom or third-party labeling tool, or choose to label images manually, to take advantage of other features, such as assigning image labeling tasks to multiple team members.

If you choose to use a labeling tool other than the Azure AI Custom Vision portal, you may need to adjust the output to match the measurement units expected by the Azure AI Custom Vision API. Bounding boxes are defined by four values that represent the left (X) and top (Y) coordinates of the top-left corner of the bounding box, and the width and height of the bounding box. These values are expressed as *proportional* values relative to the source image size. For example, consider this bounding box:

- Left: 0.1
- Top: 0.5
- Width: 0.5
- Height: 0.25

This defines a box in which the left is located 0.1 (one tenth) from the left edge of the image, and the top is 0.5 (half the image height) from the top. The box is half the width and a quarter of the height of the overall image.

The following image shows labeling information in JSON format for objects in an image.



4. Develop an object detection client application

<https://learn.microsoft.com/en-us/training/modules/detect-objects-images/4-use-trained-detector>

Develop an object detection client application

Completed

- 5 minutes

After you've trained an object detection model, you can use the Azure AI Custom Vision SDK to develop a client application that submits new images to be analyzed.

```
from msrest.authentication import ApiKeyCredentials
from azure.cognitiveservices.vision.customvision.prediction import CustomVisionPredictionClient

# Authenticate a client for the prediction API
credentials = ApiKeyCredentials(in_headers={"Prediction-key": "<YOUR_PREDICTION_RESOURCE_KEY>"})
prediction_client = CustomVisionPredictionClient(endpoint="<YOUR_PREDICTION_RESOURCE_ENDPOINT>",
                                                 credentials=credentials)

# Get classification predictions for an image
image_data = open("<PATH_TO_IMAGE_FILE>", "rb").read()
results = prediction_client.detect_image("<YOUR_PROJECT_ID>",
                                         "<YOUR_PUBLISHED_MODEL_NAME>",
                                         image_data)

# Process predictions
for prediction in results.predictions:
    if prediction.probability > 0.5:
        left = prediction.bounding_box.left
        top = prediction.bounding_box.top
        height = prediction.bounding_box.height
        width = prediction.bounding_box.width
        print(f"{prediction.tag_name} ({prediction.probability})")
        print(f" Left:{left}, Top:{top}, Height:{height}, Width:{width}")
```

```

using System;
using System.IO;
using Microsoft.Azure.CognitiveServices.Vision.CustomVision.Prediction;

// Authenticate a client for the prediction API
CustomVisionPredictionClient prediction_client = new CustomVisionPredictionClient(new ApiKeyService
{
    Endpoint = "<YOUR_PREDICTION_RESOURCE_ENDPOINT>"
};

// Get classification predictions for an image
MemoryStream image_data = new MemoryStream(File.ReadAllBytes("<PATH_TO_IMAGE_FILE>"));
var result = prediction_client.DetectImage("<YOUR_PROJECT_ID>",
    "<YOUR_PUBLISHED_MODEL_NAME>",
    image_data);

// Process predictions
foreach (var prediction in result.Predictions)
{
    if (prediction.Probability > 0.5)
    {
        var left = prediction.BoundingBox.Left;
        var top = prediction.BoundingBox.Top;
        var height = prediction.BoundingBox.Height;
        var width = prediction.BoundingBox.Width;
        Console.WriteLine($"{predictionTagName} ({predictionProbability})");
        Console.WriteLine($" Left:{left}, Top:{top}, Height:{height}, Width:{width}");
    }
}

```

5. Exercise - Detect objects in images

<https://learn.microsoft.com/en-us/training/modules/detect-objects-images/5-exercise-custom-vision>

Exercise - Detect objects in images

Completed

- 45 minutes

If you have an Azure subscription, you can use Azure AI Custom Vision to create a custom object detection model for yourself.

In this exercise, you'll train an object detection model and test it from a client application.

Note

If you don't have an Azure subscription, and you want to explore Azure AI Studio, you can [sign up for an account](#), which includes credits for the first 30 days.

Launch the exercise and follow the instructions.

[Launch Exercise](#)

Tip

After completing the exercise, if you've finished exploring Foundry Tools, delete the Azure resources that you created during the exercise.

6. Module assessment

<https://learn.microsoft.com/en-us/training/modules/detect-objects-images/6-knowledge-check>

Module assessment

Completed

- 3 minutes

7. Summary

<https://learn.microsoft.com/en-us/training/modules/detect-objects-images/7-summary>

Summary

Completed

- 1 minute

In this module, you have learned how to use the **Azure AI Custom Vision** service to create object detection models.

Tip

To find out more about the Azure AI Custom Vision service, see the [Azure AI Custom Vision documentation](#).