

Quickstart: Build an object detector with the Custom Vision website

Article • 07/18/2023

In this quickstart, you'll learn how to use the Custom Vision website to create an object detector model. Once you build a model, you can test it with new images and integrate it into your own image recognition app.

If you don't have an Azure subscription, create a [free account](#) before you begin.

Prerequisites

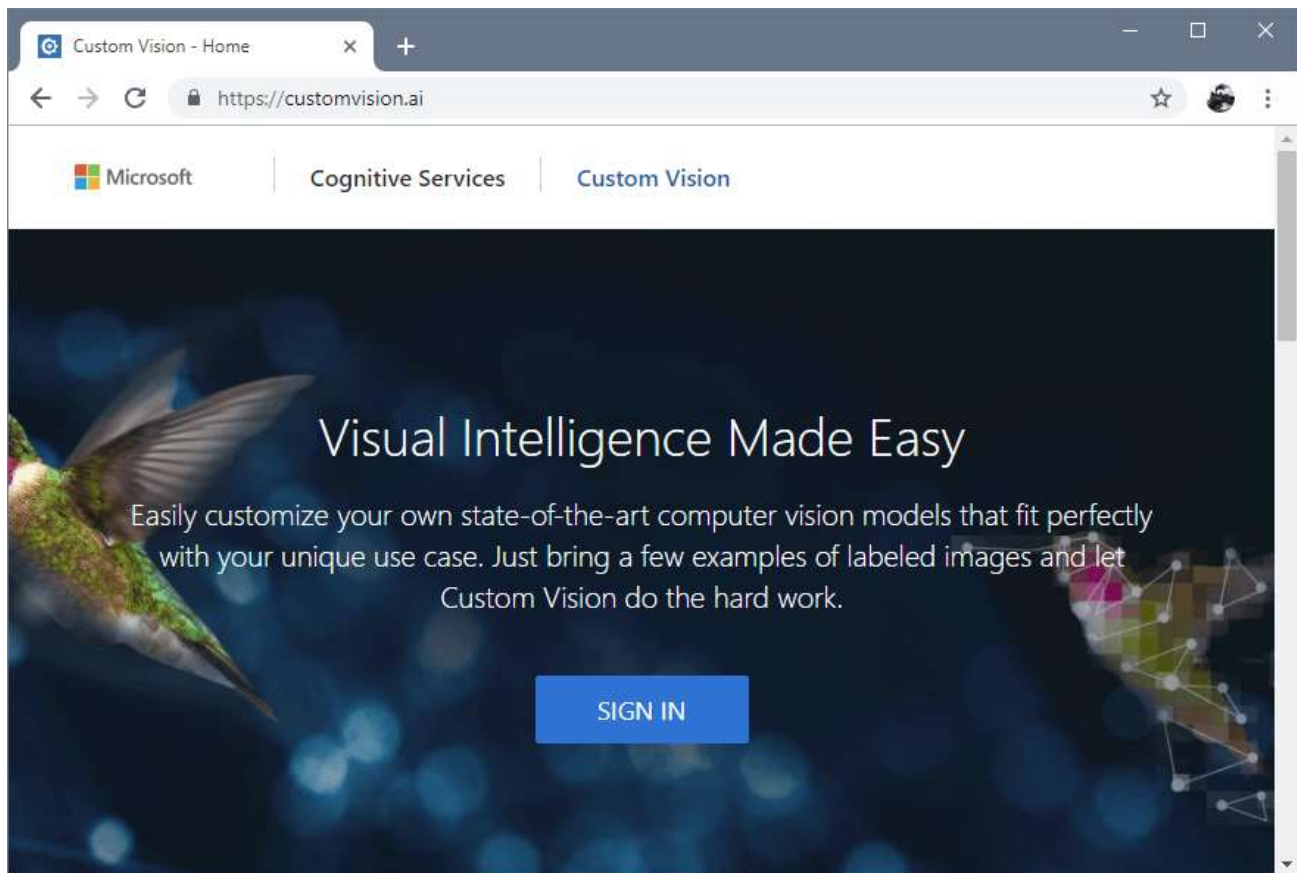
- A set of images with which to train your detector model. You can use the set of [sample images](#) on GitHub. Or, you can choose your own images using the tips below.
- A [supported web browser](#)

Create Custom Vision resources

To use the Custom Vision Service you will need to create Custom Vision Training and Prediction resources in Azure. To do so in the Azure portal, fill out the dialog window on the [Create Custom Vision](#) page to create both a Training and Prediction resource.

Create a new project

In your web browser, navigate to the [Custom Vision web page](#) and select **Sign in**. Sign in with the same account you used to sign in to the Azure portal.



1. To create your first project, select **New Project**. The **Create new project** dialog box will appear.

Create new project ✕

Name*

Description

Resource [create new](#)

[Manage Resource Permissions](#)

Project Types ⓘ

☐ Classification

☒ Object Detection

Domains:

☒ General [A1]

☐ General

☐ Logo

☐ Products on Shelves

☐ General (compact) [S1]

☐ General (compact)

Pick the domain closest to your scenario. Compact domains are lightweight models that can be exported to iOS/Android and other platforms. [Learn More](#)

CancelCreate project

2. Enter a name and a description for the project. Then select your Custom Vision Training Resource. If your signed-in account is associated with an Azure account, the

Resource dropdown will display all of your compatible Azure resources.

ⓘ Note

If no resource is available, please confirm that you have logged into **customvision.ai** with the same account as you used to sign in to the **Azure portal**. Also, please confirm you have selected the same "Directory" in the Custom Vision website as the directory in the Azure portal where your Custom Vision resources are located. In both sites, you may select your directory from the drop down account menu at the top right corner of the screen.

3. Under
4. Select **Object Detection** under **Project Types**.
5. Next, select one of the available domains. Each domain optimizes the detector for specific types of images, as described in the following table. You can change the domain later if you want to.

 **Expand table**

Domain	Purpose
General	Optimized for a broad range of object detection tasks. If none of the other domains are appropriate, or if you're unsure about which domain to choose, select the General domain.
Logo	Optimized for finding brand logos in images.
Products on shelves	Optimized for detecting and classifying products on shelves.
Compact domains	Optimized for the constraints of real-time object detection on mobile devices. The models generated by compact domains can be exported to run locally.

6. Finally, select **Create project**.

Choose training images

As a minimum, we recommend you use at least 30 images per tag in the initial training set. You'll also want to collect a few extra images to test your model once it's trained.

In order to train your model effectively, use images with visual variety. Select images that vary by:

- camera angle
- lighting
- background
- visual style
- individual/grouped subject(s)
- size
- type

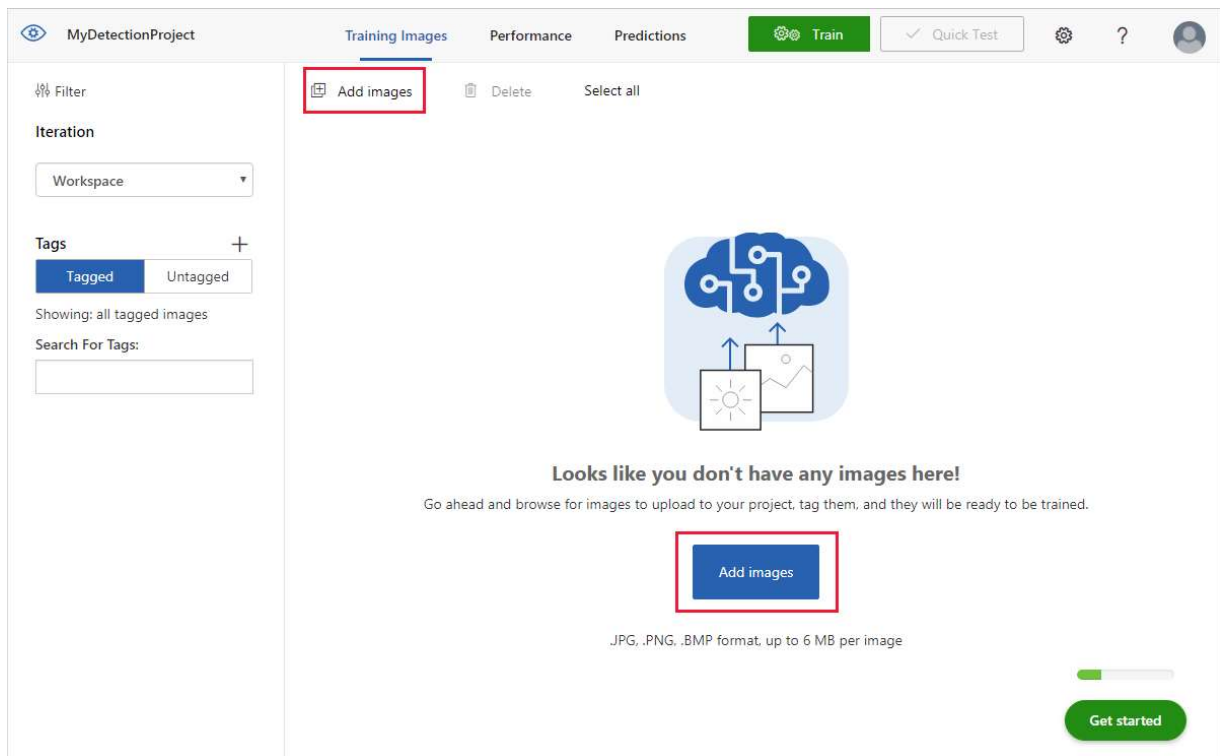
Additionally, make sure all of your training images meet the following criteria:

- .jpg, .png, .bmp, or .gif format
- no greater than 6MB in size (4MB for prediction images)
- no less than 256 pixels on the shortest edge; any images shorter than this will be automatically scaled up by the Custom Vision Service

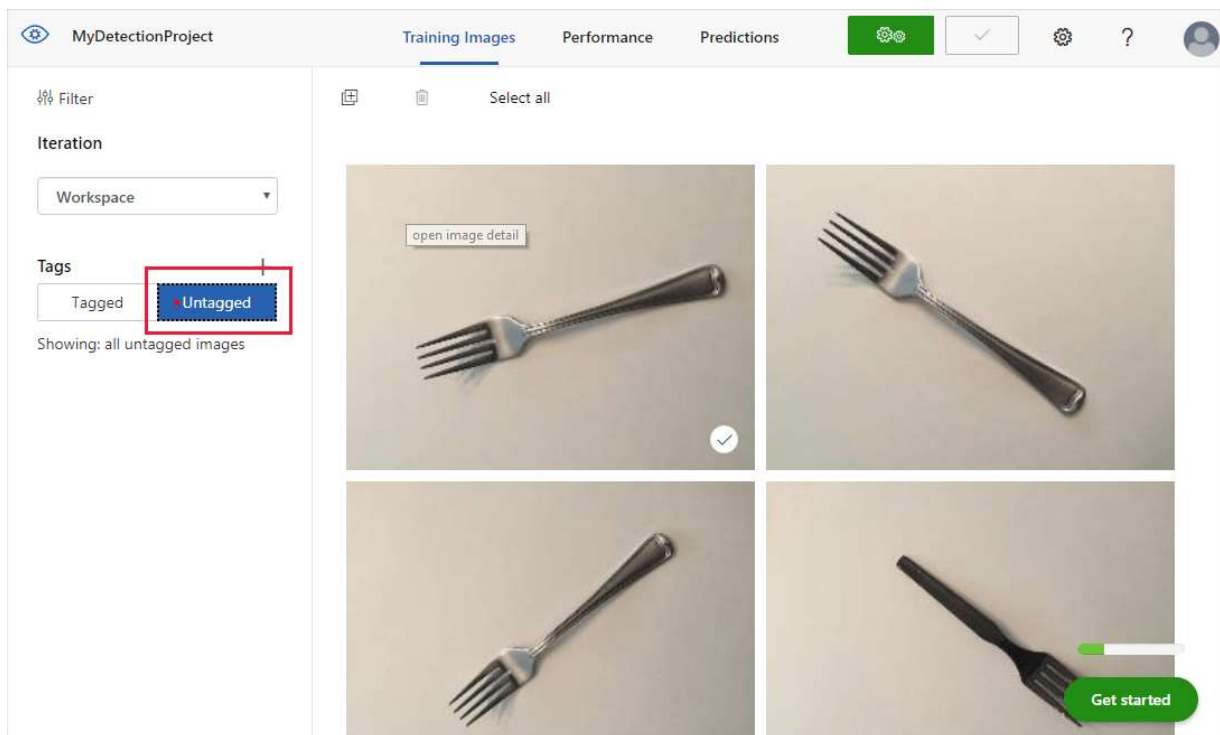
Upload and tag images

In this section, you'll upload and manually tag images to help train the detector.

1. To add images, select **Add images** and then select **Browse local files**. Select **Open** to upload the images.

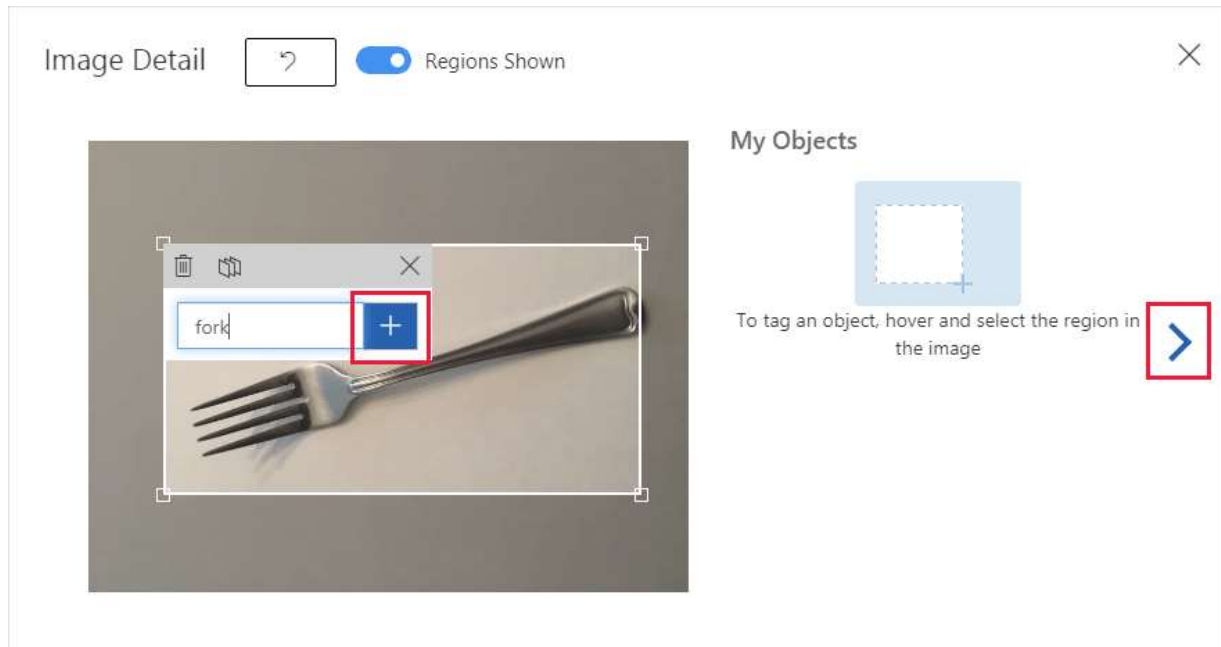


2. You'll see your uploaded images in the **Untagged** section of the UI. The next step is to manually tag the objects that you want the detector to learn to recognize. Select the first image to open the tagging dialog window.



3. Select and drag a rectangle around the object in your image. Then, enter a new tag name with the + button, or select an existing tag from the drop-down list. It's important to tag every instance of the object(s) you want to detect, because the detector uses the untagged background area as a negative example in training. When

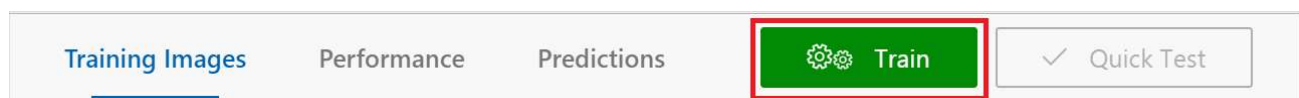
you're done tagging, select the arrow on the right to save your tags and move on to the next image.



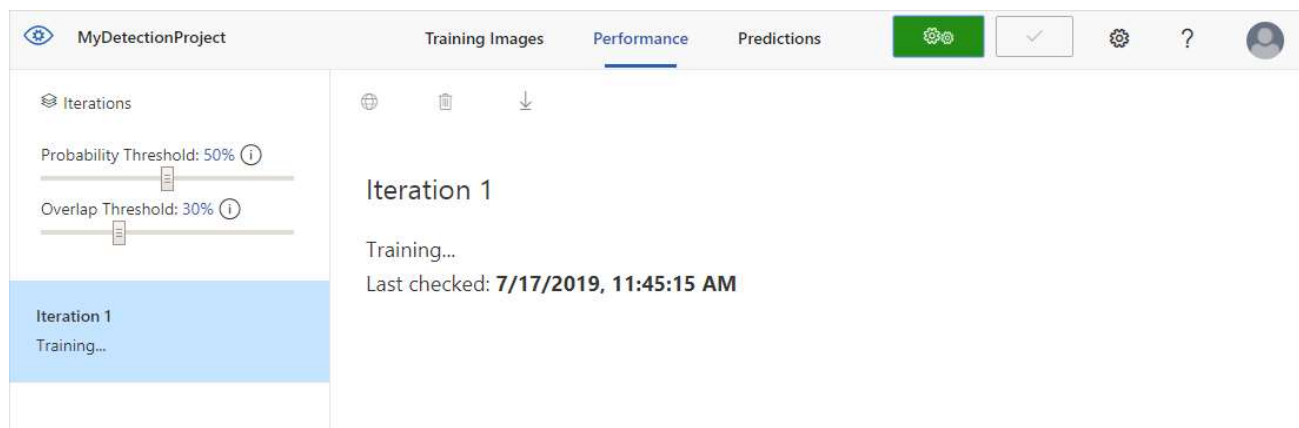
To upload another set of images, return to the top of this section and repeat the steps.

Train the detector

To train the detector model, select the **Train** button. The detector uses all of the current images and their tags to create a model that identifies each tagged object. This process can take several minutes.



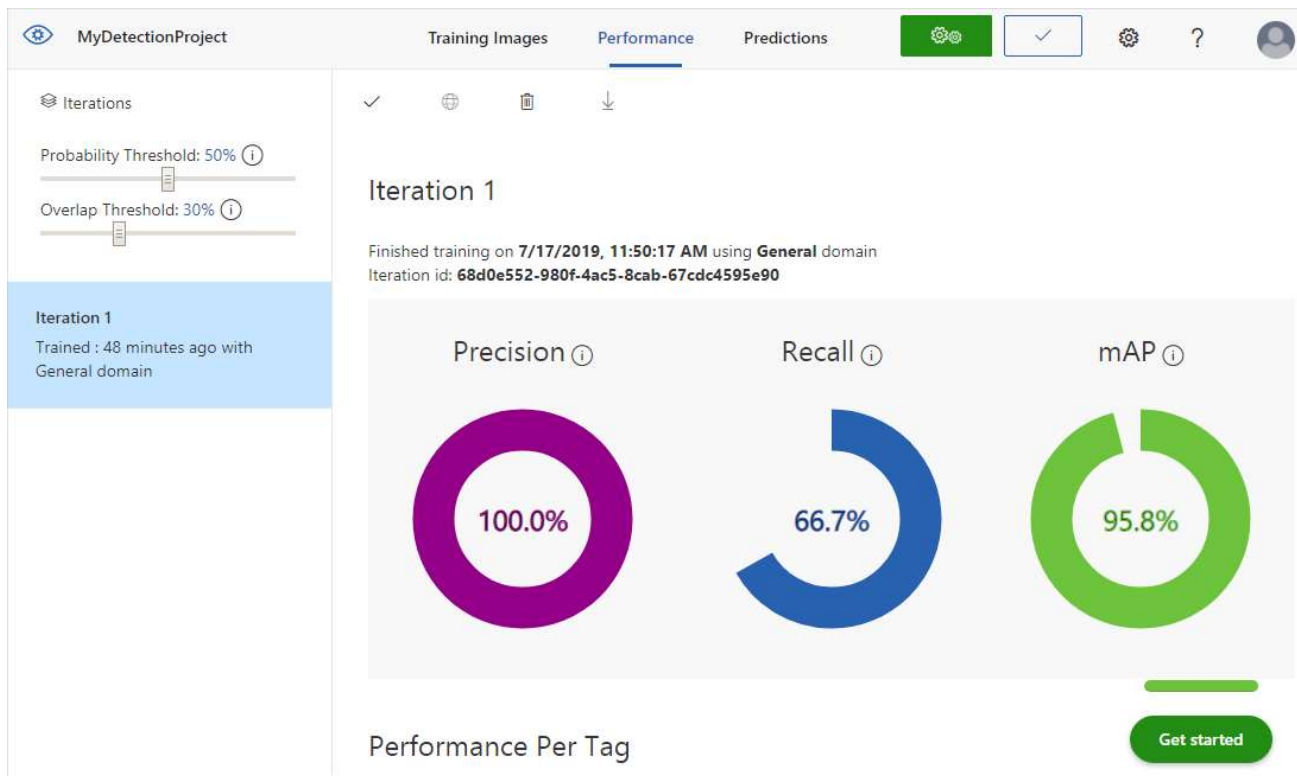
The training process should only take a few minutes. During this time, information about the training process is displayed in the **Performance** tab.



Evaluate the detector

After training has completed, the model's performance is calculated and displayed. The Custom Vision service uses the images that you submitted for training to calculate precision, recall, and mean average precision. Precision and recall are two different measurements of the effectiveness of a detector:

- **Precision** indicates the fraction of identified classifications that were correct. For example, if the model identified 100 images as dogs, and 99 of them were actually of dogs, then the precision would be 99%.
- **Recall** indicates the fraction of actual classifications that were correctly identified. For example, if there were actually 100 images of apples, and the model identified 80 as apples, the recall would be 80%.
- **Mean average precision** is the average value of the average precision (AP). AP is the area under the precision/recall curve (precision plotted against recall for each prediction made).



Probability threshold

Note the **Probability Threshold** slider on the left pane of the **Performance** tab. This is the level of confidence that a prediction needs to have in order to be considered correct (for the purposes of calculating precision and recall).

When you interpret prediction calls with a high probability threshold, they tend to return results with high precision at the expense of recall—the detected classifications are correct, but many remain undetected. A low probability threshold does the opposite—most of the actual classifications are detected, but there are more false positives within that set. With this in mind, you should set the probability threshold according to the specific needs of your project. Later, when you're receiving prediction results on the client side, you should use the same probability threshold value as you used here.

Overlap threshold

The **Overlap Threshold** slider deals with how correct an object prediction must be to be considered "correct" in training. It sets the minimum allowed overlap between the predicted object's bounding box and the actual user-entered bounding box. If the bounding boxes don't overlap to this degree, the prediction won't be considered correct.

Manage training iterations

Each time you train your detector, you create a new *iteration* with its own updated performance metrics. You can view all of your iterations in the left pane of the **Performance** tab. In the left pane you'll also find the **Delete** button, which you can use to delete an iteration if it's obsolete. When you delete an iteration, you delete any images that are uniquely associated with it.

See [Use your model with the prediction API](#) to learn how to access your trained models programmatically.

Next steps

In this quickstart, you learned how to create and train an object detector model using the Custom Vision website. Next, get more information on the iterative process of improving your model.

Test and retrain a model

- [What is Custom Vision?](#)