

Classify images

1. Introduction

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

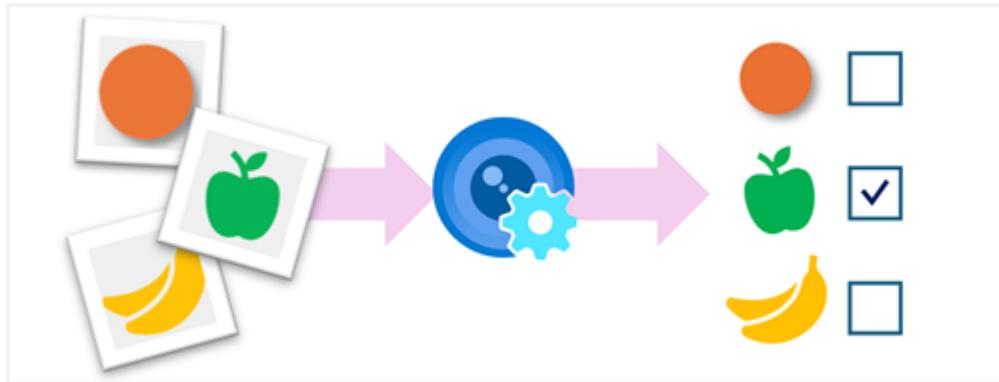
Introduction

Completed

- 1 minute

Image classification is a common computer vision problem that requires software to analyze an image and categorize (or *classify*) it.

For example, an unattended checkout system in a grocery store might use a camera to scan each item a customer adds to their cart, and use image classification to identify the product in the image.



In this module, you'll learn how the **Azure AI Custom Vision** service enables you to build your own computer vision models for image classification.

2. Azure AI Custom Vision

Azure AI Custom Vision

Completed

- 5 minutes

The **Azure AI Custom Vision** service enables you to build your own computer vision models for *image classification or object detection*.

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

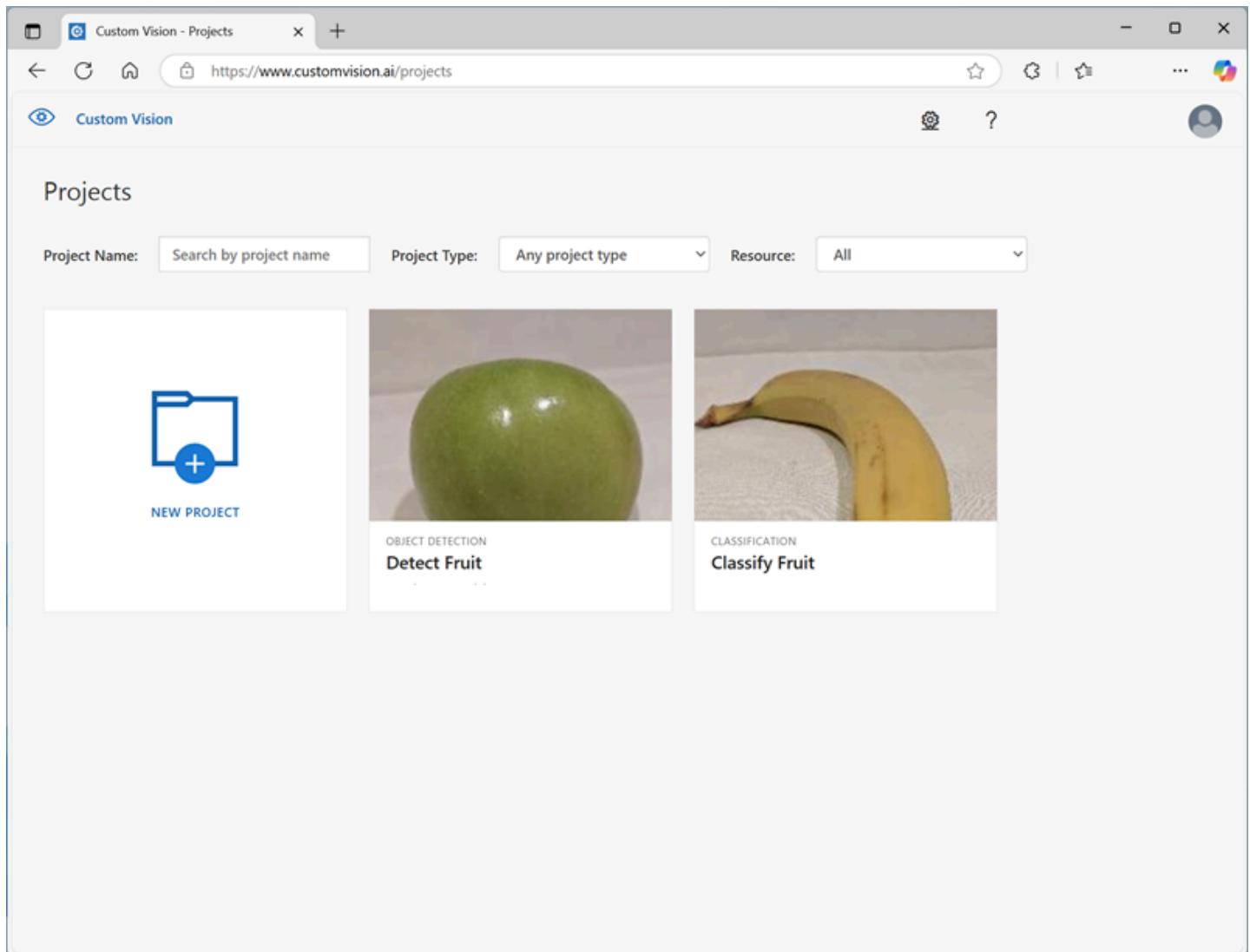
- An **Azure AI Custom Vision *training*** resource - used to train a custom model based on your own training images.
- An **Azure AI Custom Vision *prediction*** resource - used to generate predictions from new images based on your trained model.

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 image classification model

<https://learn.microsoft.com/en-us/training/modules/classify-images/3-understand-image-classification>

Train an image classification model

Completed

- 3 minutes

Image classification is a computer vision technique in which a model is trained to predict a class label for an image based on its contents. Usually, the class label relates to the main *subject* of the image.

For example, the following images have been classified based on the type of fruit they contain.



Apple



Banana



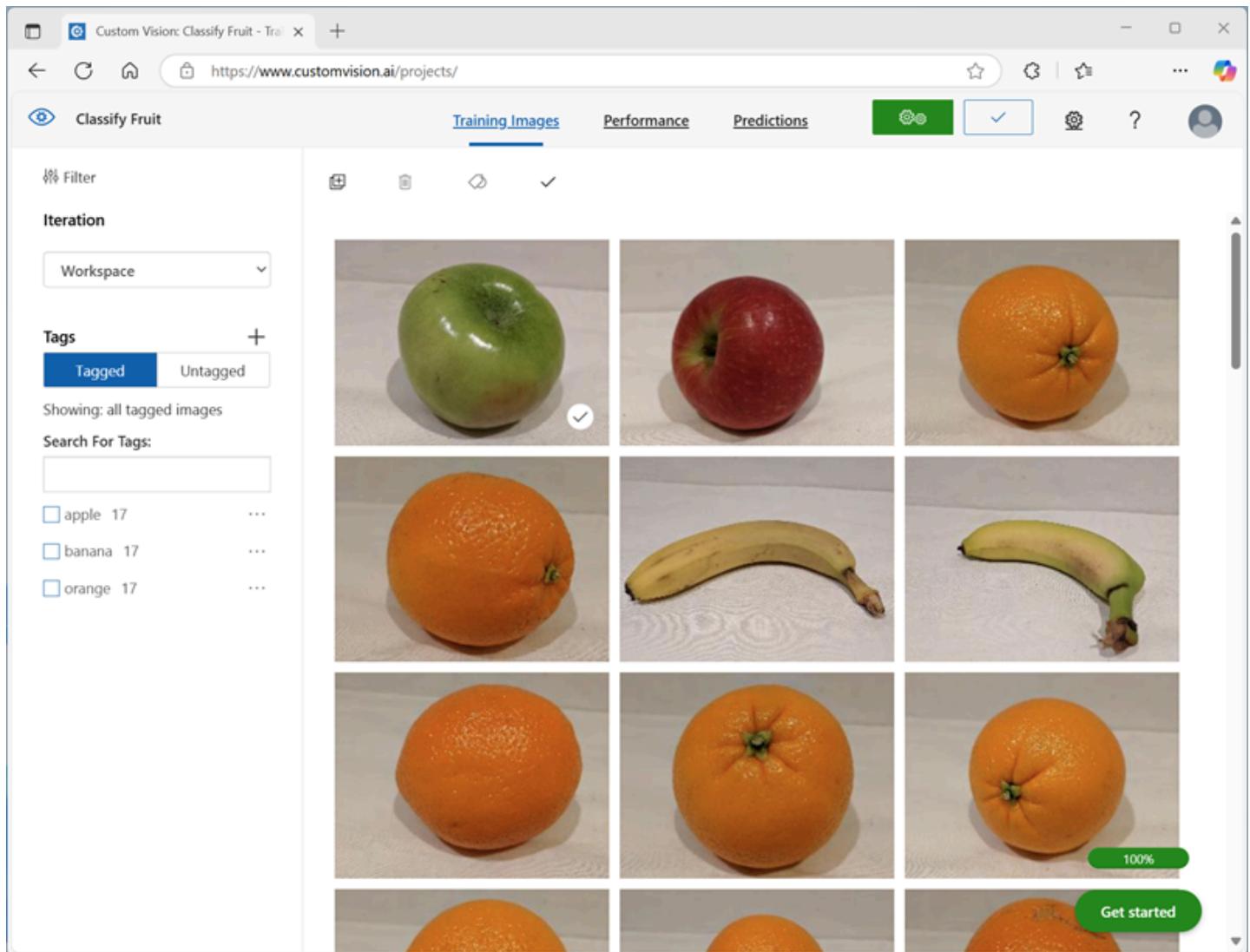
Orange

Models can be trained for multiclass classification (in other words, there are multiple classes, but each image can belong to only one class) or multilabel classification (in other words, an image might be associated with multiple labels).

Training an image classification model

To train an image classification model with the Azure AI Custom Vision service, you can use the Azure AI Custom Vision portal, the Azure AI Custom Vision REST API or SDK, or a combination of both approaches.

In most cases, you'll typically use the Azure AI Custom Vision portal to train your model.



The portal provides a graphical interface that you can use to:

1. Create an image classification project for your model and associate it with a training resource.
2. Upload images, assigning class label tags to them.
3. Review and edit tagged images.
4. Train and evaluate a classification model.
5. Test a trained model.
6. Publish a trained model to a prediction resource.

The REST API and SDKs enable you to perform the same tasks by writing code, which is useful if you need to automate model training and publishing as part of a DevOps process.

4. Create an image classification client application

<https://learn.microsoft.com/en-us/training/modules/classify-images/4-use-trained-image-classifier>

Create an image classification client application

Completed

- 3 minutes

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

```
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.classify_image("<YOUR_PROJECT_ID>",
                                           "<YOUR_PUBLISHED_MODEL_NAME>",
                                           image_data)

# Process predictions
for prediction in results.predictions:
    if prediction.probability > 0.5:
        print(image, ': {} {:.0%}'.format(prediction.tag_name, prediction.probability))
```

```
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 ApiService
{
    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.ClassifyImage("<YOUR_PROJECT_ID>",
                                             "<YOUR_PUBLISHED_MODEL_NAME>",
                                             image_data);

// Process predictions
foreach (var prediction in result.Predictions)
```

```
{  
    if (prediction.Probability > 0.5)  
    {  
        Console.WriteLine($"{{prediction.TagName}} ({prediction.Probability}})");  
    }  
}
```

5. Exercise - Classify images

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

Exercise - Classify images

Completed

- 45 minutes

Now it's your turn to try using the Azure AI Custom Vision service.

In this exercise, you train and publish a custom image classification model, and use the Azure AI Custom Vision SDK to test it.

Note

To complete this lab, you need an [Azure subscription](#) in which you have administrative access.

Launch the exercise and follow the instructions.

[Launch Exercise](#)

When you finish the exercise, don't forget to come back and complete the knowledge check to earn points for completing this module!

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/classify-images/6-knowledge-check>

Module assessment

Completed

- 3 minutes

7. Summary

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

Summary

Completed

- 3 minutes

In this module, you learned how to use the **Azure AI Custom Vision** service to build your own custom vision models for image classification.

Tip

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