2.2 - Image classification with custom Azure AI Vision models

- Overview
- Introduction
- Understand custom model types
  - Image classification
  - Object detection
  - Product recognition
- Create a custom project
  - Components of a custom Vision project
  - COCO files
  - Creating your dataset
- Label and train a custom model
  - Labeling your training images
  - Training your model
- Exercise - Classify images with an Azure AI Vision custom model
  - Provision Azure resources
  - Create a custom model training project
  - Test your custom model
- Knowledge Check
- Summary

Overview

Computer vision is an area of artificial intelligence that deals with visual perception. Azure AI Vision includes multiple services that support common computer vision scenarios.

Introduction

Custom models in Azure AI Vision allow you to train an AI model to classify images or detect objects in images.

Image classification is a common computer vision problem that requires software to analyze an image in order to categorize (or *classify*) it. In this module, you'll learn how the **Azure AI Vision** service enables you to build your own custom vision models for image classification.

Object detection is another common computer vision problem that requires software to identify the location of specific classes of object in an image. **Creating an object detection project follows the same pattern as an image classification project, from creation through labeling and training.** This module covers concepts and considerations for object detection, however the exercise focuses on classification.

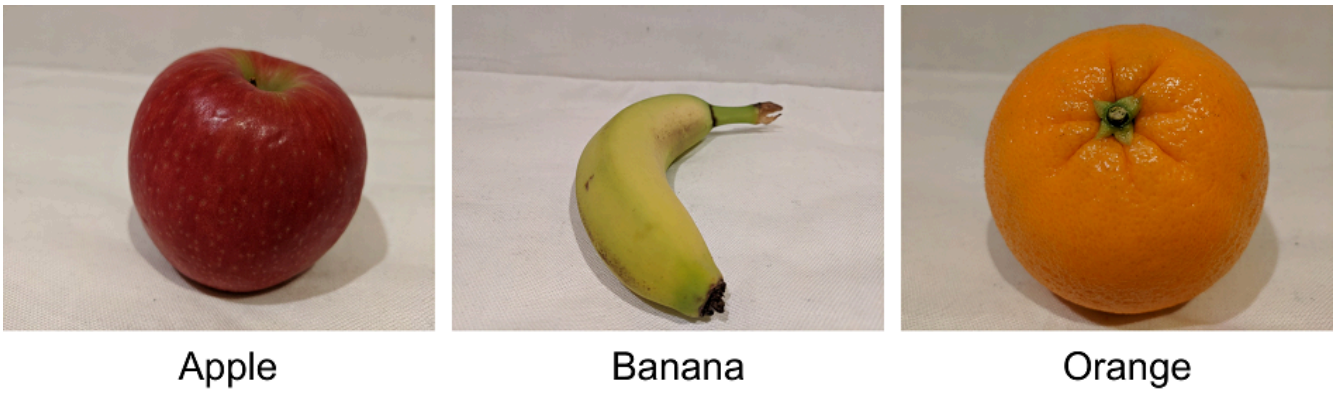
Understand custom model types

Custom Azure AI Vision models have different functionality based on the *type*. The types of custom models include **Image classification**, **Object detection**, and **Product recognition**.

Image classification

Image classification is a computer vision feature where a model is trained to predict a label for an image based on the contents of the entire image. Usually the class label relates to the main *subject* of the image, however individual use cases may vary.

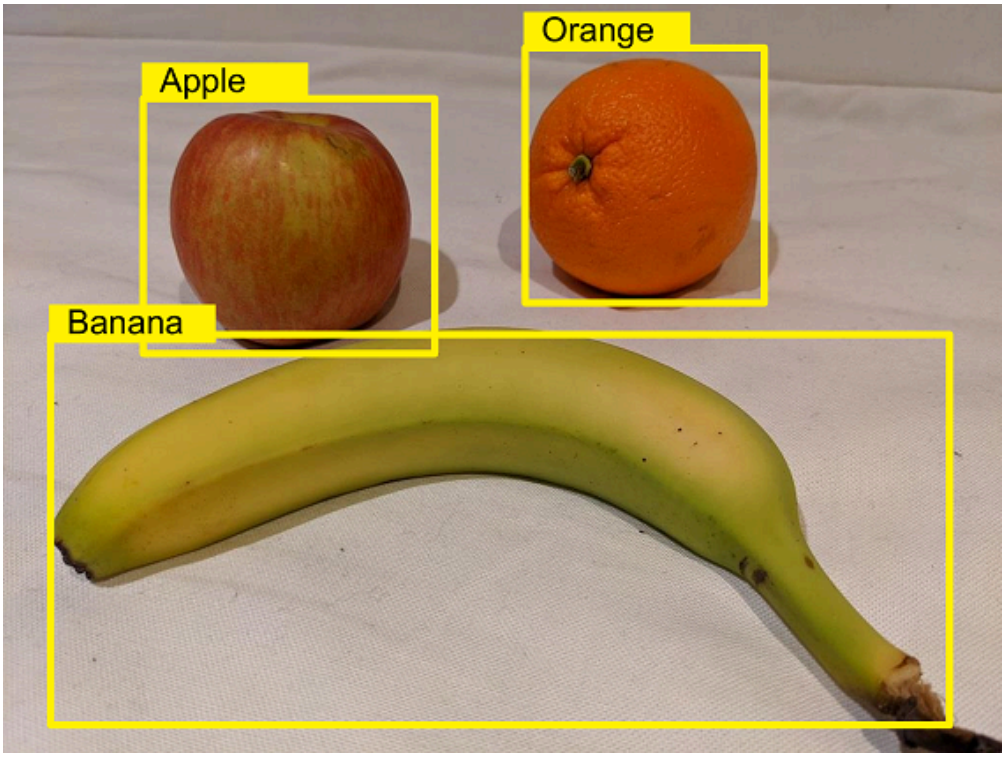
For example, the following images are classified based on the type of fruit they contain.



Models can be trained for multi-class classification (where there are multiple classes, but each image can belong to only one class) or multi-label classification (where an image might be associated with multiple labels).

Object detection

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. For example, an AI enabled checkout system in a grocery store might need to identify the type and location of items being purchased.



There are two components to object detection:

- The class label of each object detected in the image. For example, you might predict that an image contains one apple and two oranges.
- The location of each object within the image, indicated as coordinates of a bounding box that encloses the object.

Product recognition

Product recognition works the same way object detection does, but with **improved accuracy for product labels and brand names**. The predictions for product recognition have both the class label and location, enabling you to know where in the image the product is.

Create a custom project

To create a custom Azure AI Vision model, you first need an **Azure AI Services resource** (or an **Azure AI Vision resource**). Once that resource is deployed to your subscription, you need to create a custom project.

Components of a custom Vision project

The first component of a custom project is the *dataset*. The dataset is your collection of images to use while training your model, along with the *COCO file* that defines the label information about those images. Your **dataset is stored in an Azure blob storage container**, and we cover more about the COCO file further in this unit.

Once you have your images and class labels defined, you can train your custom model. When training your model, you specify the type of model to train, which dataset to use, and your training budget (in amount of time). When the model training is complete, you can view the performance and use the model for predictions.

In most cases, the steps you follow are:

1. Create your blob storage container and upload just the training images.
2. Create the dataset for your project, and **connect it to your blob storage container**. When creating your dataset, you define what type of project it is (image classification, object detection, or product recognition).
3. Label your data in your **Azure Machine Learning Data Labeling Project**, which creates the COCO file in your blob storage container.
4. **Connect your completed COCO file for the labeled images to your dataset.**
5. Train your custom model on the dataset and labels created.
6. Verify performance and iterate if the trained performance isn't meeting expectations.

Once you're happy with the performance, the model can be used in Vision Studio or in your own application.

COCO files

A COCO file is a JSON file with a specific format that defines:

- **images**: Defines the image location in blob storage, name, width, height, and ID.
- **annotations**: Defines the classifications (or objects), including which category the image is classified as, the area, and the bounding box (if labeling for object detection).
- **categories**: Defines the ID for the named label class.

In most cases, COCO files are created by labeling your training images in an Azure Machine Learning Data Labeling Project.

If you're migrating from an old Custom Vision project, you can use the migration script to create your COCO file.

A sample COCO file looks like this:

```
{
  "images": [
    {
      "id": 1,
      "width": 1024,
      "height": 768,
      "file_name": "abc.jpg",
      "coco_url": "AmlDatastore://fruit/abc.jpg",
      "absolute_url": "https://myBlobStorage.blob.core.windows.net/fruit/abc.jpg",
      "date_captured": "<date>"
    },
    {
      "id": 2,
      "width": 1024,
      "height": 768,
      "file_name": "xyz.jpg",
      "coco_url": "AmlDatastore://fruit/xyz.jpg",
      "absolute_url": "https://myBlobStorage.blob.core.windows.net/fruit/xyz.jpg",
      "date_captured": "<date>"
    },
    <...>
  ],
  "annotations": [
    {
      "id": 1,
      "category_id": 1,
      "image_id": 1,
      "area": 0.0
    },
    {
      "id": 2,
      "category_id": 1,
      "image_id": 2,
      "area": 0.0
    },
    <...>
  ],
```

```
  "categories": [
    {
      "id": 1,
      "name": "apple"
    },
    {
      "id": 2,
      "name": "orange"
    },
    {
      "id": 3,
      "name": "banana"
    }
  ]
}
```

If you're labeling an object detection dataset, each annotation in the COCO file also contains a bounding box array with the values in the array being *Left*, *Top*, *Width*, and *Height*.

```
"bbox": [
    0.11803319477782331,
    0.41586723392402375,
    0.7765206955096307,
    0.3483334397217212
]
```

Creating your dataset

Once you have images in your blob storage container, you can create your dataset for training using either the REST API, or by using the **Vision Studio**. The REST request would be similar to the following REST call:

```
curl -X PUT https://<endpoint>/computervision/datasets/<dataset-name>?api-version=
<version>\
  -H "Content-Type: application/json" \
  -H "Ocp-Apim-Subscription-Key: <subscription-key>" \
  --data-ascii "
  {
    'annotationKind':'imageClassification',
    'annotationFileUris':['<URI>']
  }"
```

If using Vision Studio, you would navigate to the custom model tile, select your resource, and create your dataset. From there, you can open or create an Azure Machine Learning Data Labeling Project, or upload an existing COCO file. The exercise in this module walks through how to create your dataset in this way.

Using Vision Studio enables you to connect to your labeling project in Azure Machine Learning instead of specifying the COCO file in the REST request. The rest of the examples in this module

use Vision Studio, but if REST is your preferred method examples are available on the documentation pages.

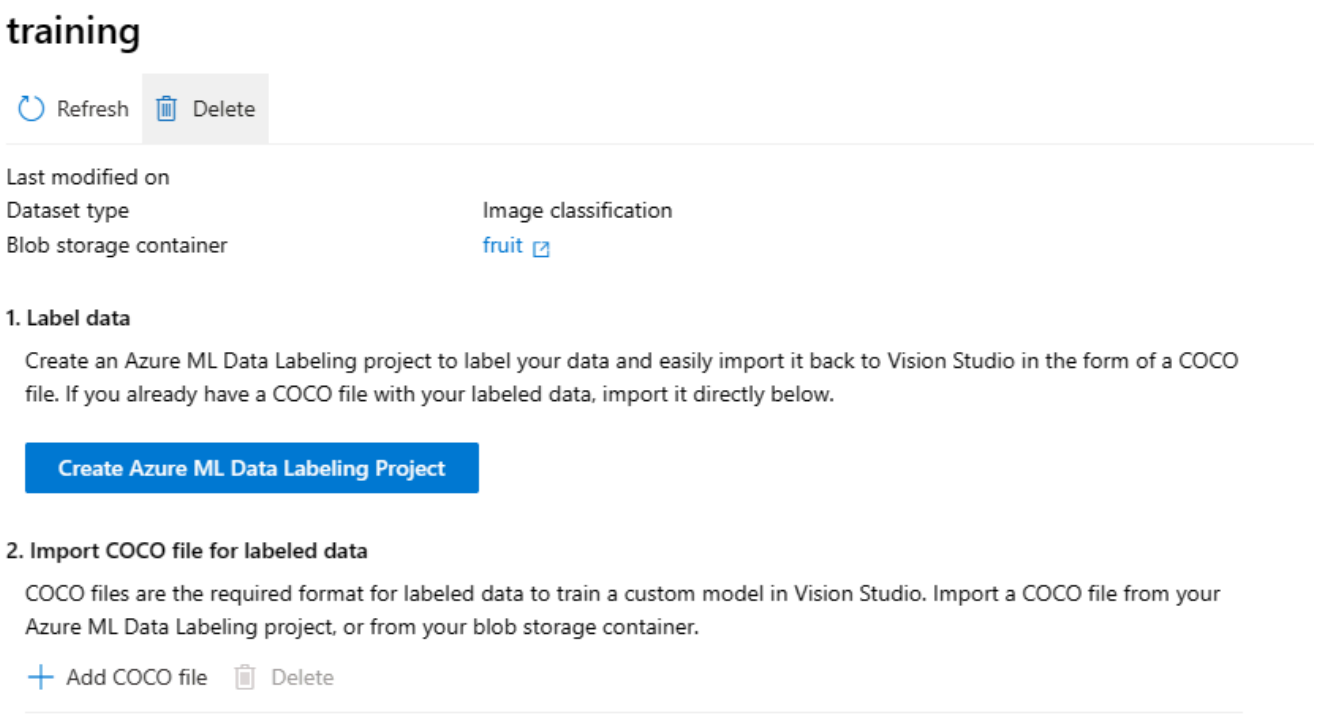
Label and train a custom model

Once you upload your images to blob storage and created your dataset, the next step is to label your images and connect the resulting COCO file. If you already have a COCO file for your training images, you can skip the labeling step.

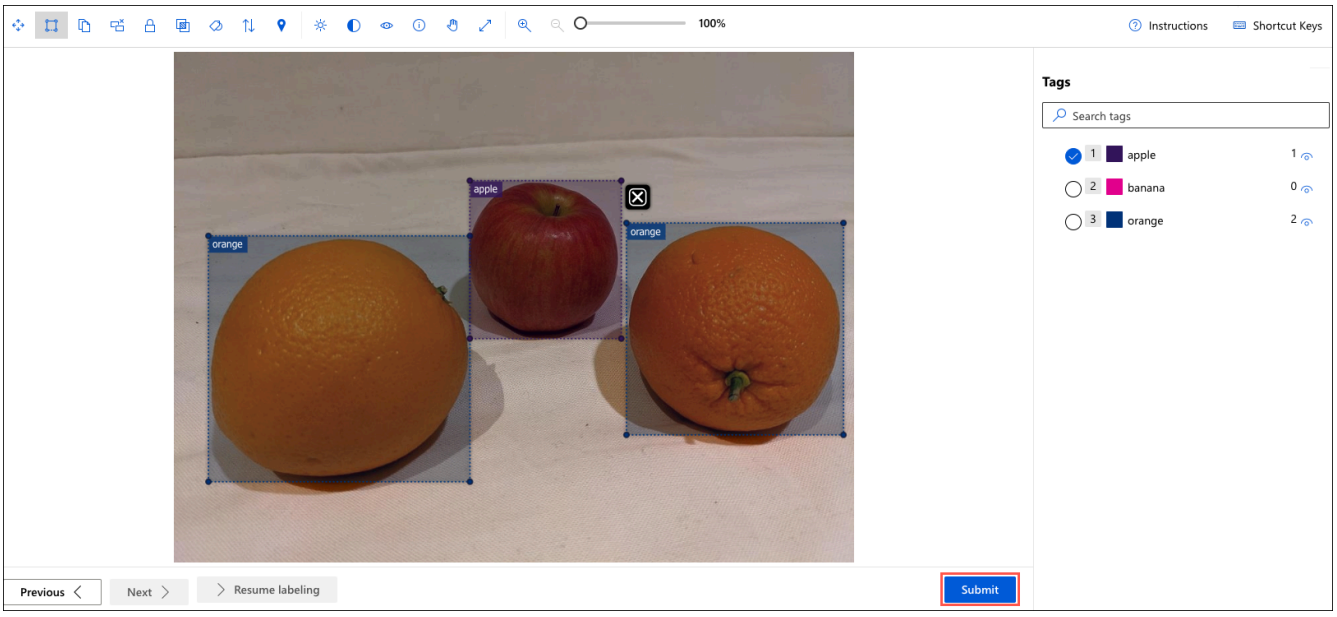
Labeling your training images

Labeling your training images is done in Azure Machine Learning studio, using the Data Labeling Project. Having complete and accurate labels for your training images greatly improves the performance of your trained model. When you label your images, be sure to accurately assign labels and completely label all instances of each class.

In your dataset within Vision Studio, create a new Azure Machine Learning Data Labeling project or connect to an existing project if you created one in Azure Machine Learning studio.



Once your project has been created, selecting that button will take you to Azure Machine Learning Studio and open up the labeling project. In Azure Machine Learning Data Labeling, you can add categories for your images or objects (such as apple, orange, banana). Once you have categories, start your project and go to the labeling tab. You need to label 3-5 images per category.



There are tools with Azure Machine Learning to help with labeling, such as *ML-assisted labeling* which takes some labels that you provide for a subset of the images and tries to label the remaining images for you. If using these features, it's important to review the labels to ensure they're accurate. If they're not accurate, your trained model's performance decreases.

When the labeling is completed and all training images are correctly classified or labeled, you can add your COCO file to your dataset directly from your Azure Machine Learning workspace.

Training your model

With all the training images labeled, the next step is training your model. When training a model select the model type, specify the dataset you want to use as training data, and indicate the training budget. The training budget is an upper bound of time for how long the training will run; the actual time used for training is often less than the specified budget.

Once your model is trained, selecting it allows you to view the performance of evaluation run. **If an evaluation dataset isn't provided when training your model, it uses the default evaluation run**. The default evaluation run takes a small set of the labeled images out of the training set, uses the trained model for predictions on that subset, and compares the predictions to the provided labels.

From the trained model page, you can trigger new evaluation runs on a different set of images or try out your own tests in Vision Studio by selecting the tab on the top of the page.

Exercise - Classify images with an Azure AI Vision custom model

Azure AI Vision enables you to train custom models to classify and detect objects with labels you specify. In this lab, we'll build a custom image classification model to classify images of fruit.

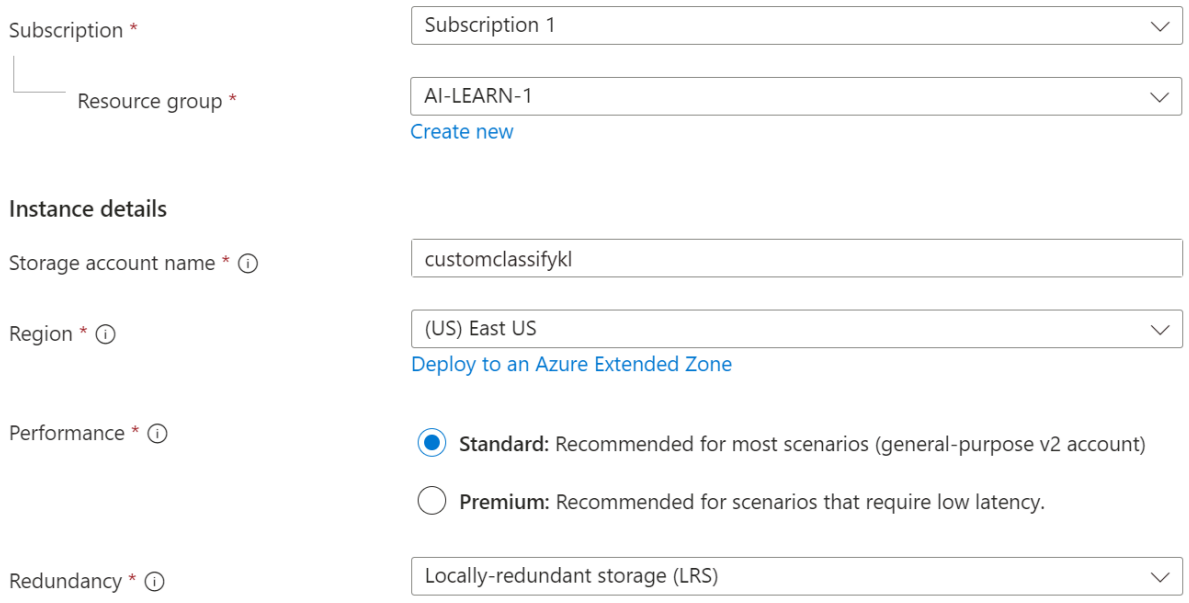
Provision Azure resources

If you don't already have one in your subscription, you'll need to provision an **Azure AI Services** resource.

1. Open the Azure portal at `https://portal.azure.com`, and sign in using the Microsoft account associated with your Azure subscription.
2. In the top search bar, search for *Azure AI services*, select **Azure AI Services**, and create an Azure AI services multi-service account resource with the following settings:
   - **Subscription**: *Your Azure subscription*
   - **Resource group**: *Choose or create a resource group (if you are using a restricted subscription, you may not have permission to create a new resource group - use the one provided)*
   - **Region**: *Choose from East US, West Europe, West US 2**
   - **Name**: *Enter a unique name*
   - **Pricing tier**: Standard S0
     *Azure AI Vision 4.0 custom model tags are currently only available in these regions.
3. Select the required checkboxes and create the resource.

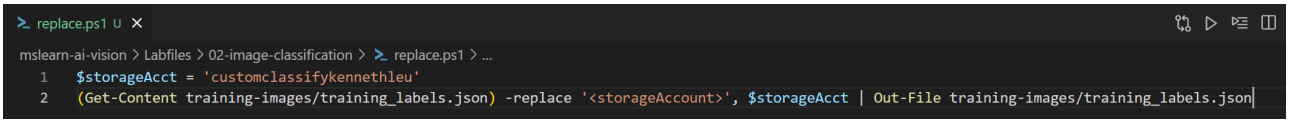
We also need a storage account to store the training images. Azure Storage is a Microsoft-managed service providing cloud storage that is highly available, secure, durable, scalable, and redundant.

1. In Azure portal, search for and select **Storage accounts**, and create a new storage account with the following settings:
   - **Subscription**: *Your Azure subscription*
   - **Resource Group**: *Choose the same resource group you created your Azure AI Service resource in*
   - **Storage Account Name**: customclassifySUFFIX
     - *note: replace the `SUFFIX` token with your initials or another value to ensure the resource name is globally unique.*
   - **Region**: *Choose the same region you used for your Azure AI Service resource*
   - **Performance**: Standard
   - **Redundancy**: Locally-redundant storage (LRS)

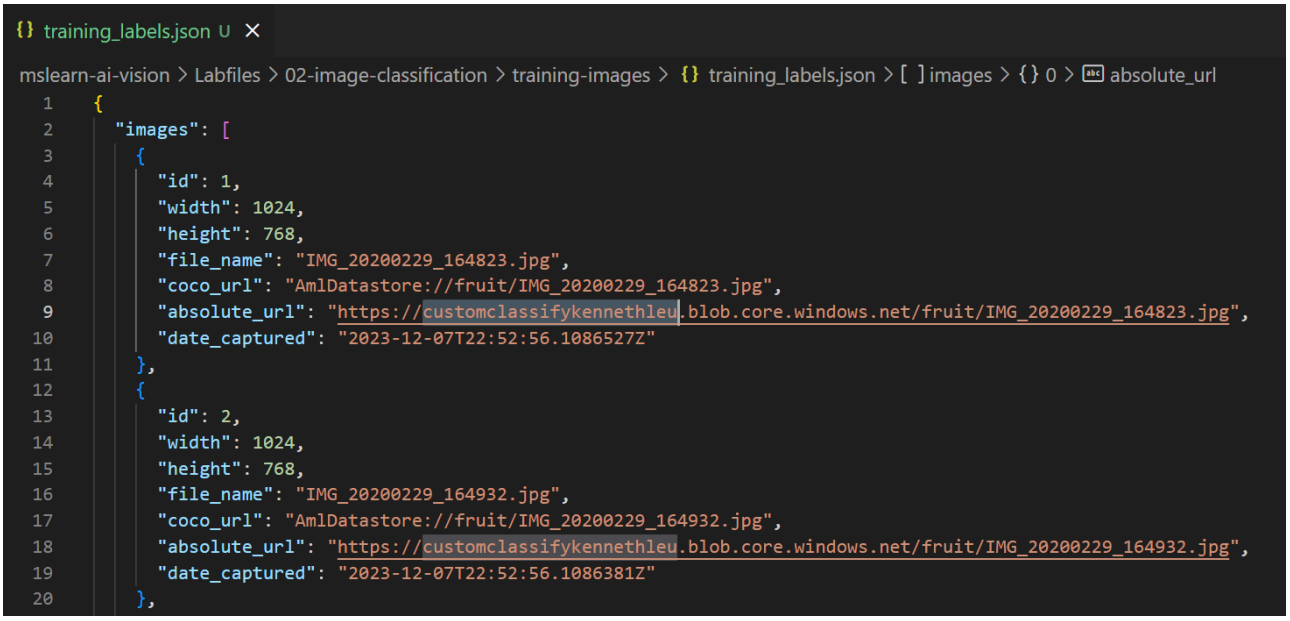


2. While your storage account is being created, go to Visual Studio Code, and expand the **Labfiles/02-image-classification** folder.
3. In that folder, select **replace.ps1** and review the code. You'll see that it replaces the name of your storage account for the placeholder in a JSON file (the COCO file) we use in a later step. Replace the

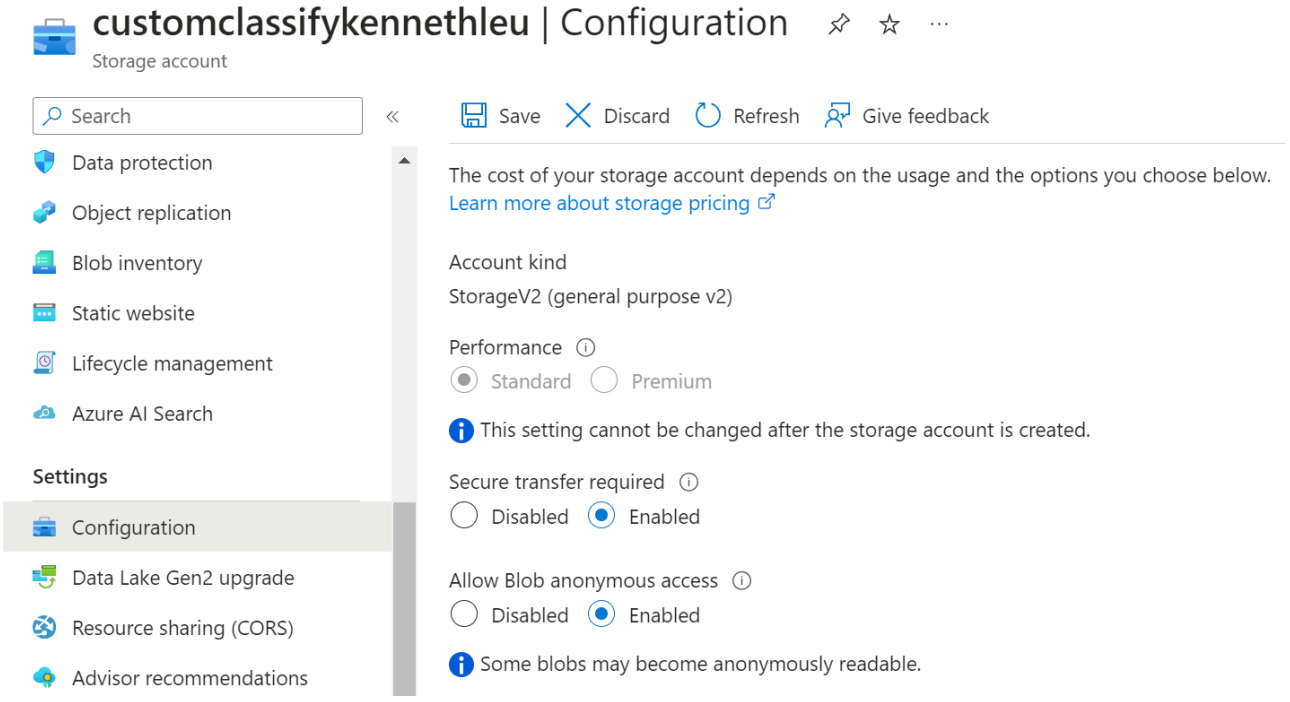
placeholder *in the first line only* of the file with the name of your storage account. Save the file.



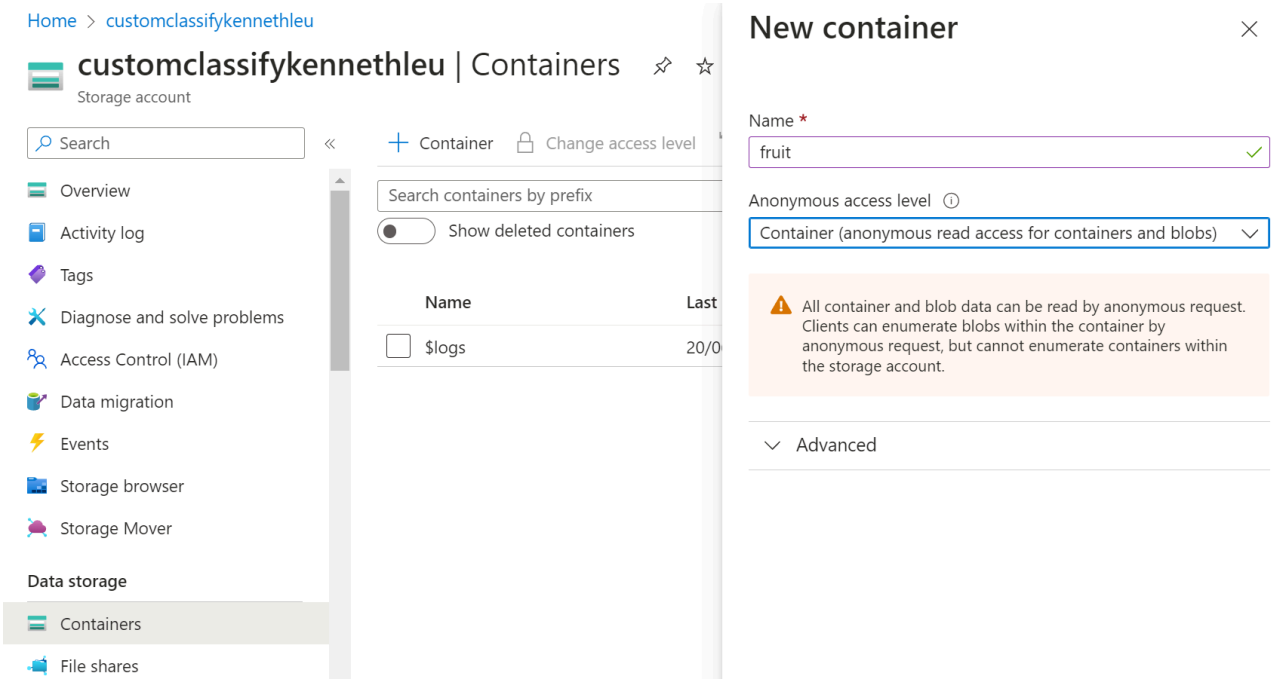
4. Right-click on the **02-image-classification** folder and open an Integrated Terminal. Run the following command in Powershell to set the storage account name for the COCO file: `./replace.ps1`
5. You can review the COCO file to ensure your storage account name is there. Select **training-images/training_labels.json** and view the first few entries. In the *absolute_url* field, you should see something similar to *"https://myStorage.blob.core.windows.net/fruit/…*. If you don't see the change expected, make sure you updated only the first placeholder in the PowerShell script.



6. Close both the JSON and PowerShell file, and go back to your browser window.
7. Your storage account should be complete. Go to your storage account.
8. Enable public access on the storage account. In the left pane, navigate to **Configuration** in the **Settings** group, and enable *Allow Blob anonymous access*. Select **Save**

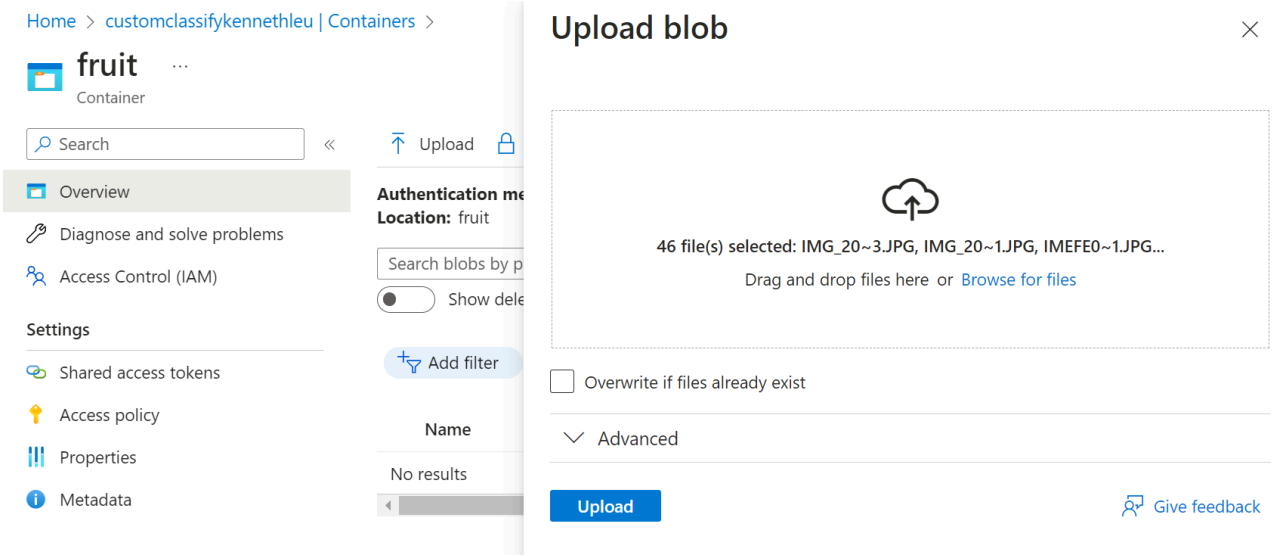


9. In the left pane, select **Containers** and create a new container named `fruit`, and set **Anonymous access level** to *Container (anonymous read access for containers and blobs)*.



> **Note**: If the **Anonymous access level** is disabled, refresh the browser page.
>
> **Azure Storage Containers** are part of the Azure Blob Storage service (which is optimized for storing large amounts of unstructured data). A storage account can contain an unlimited number of containers, and a container can store an unlimited number of blobs. An Azure Storage container is similar to an AWS S3 bucket.

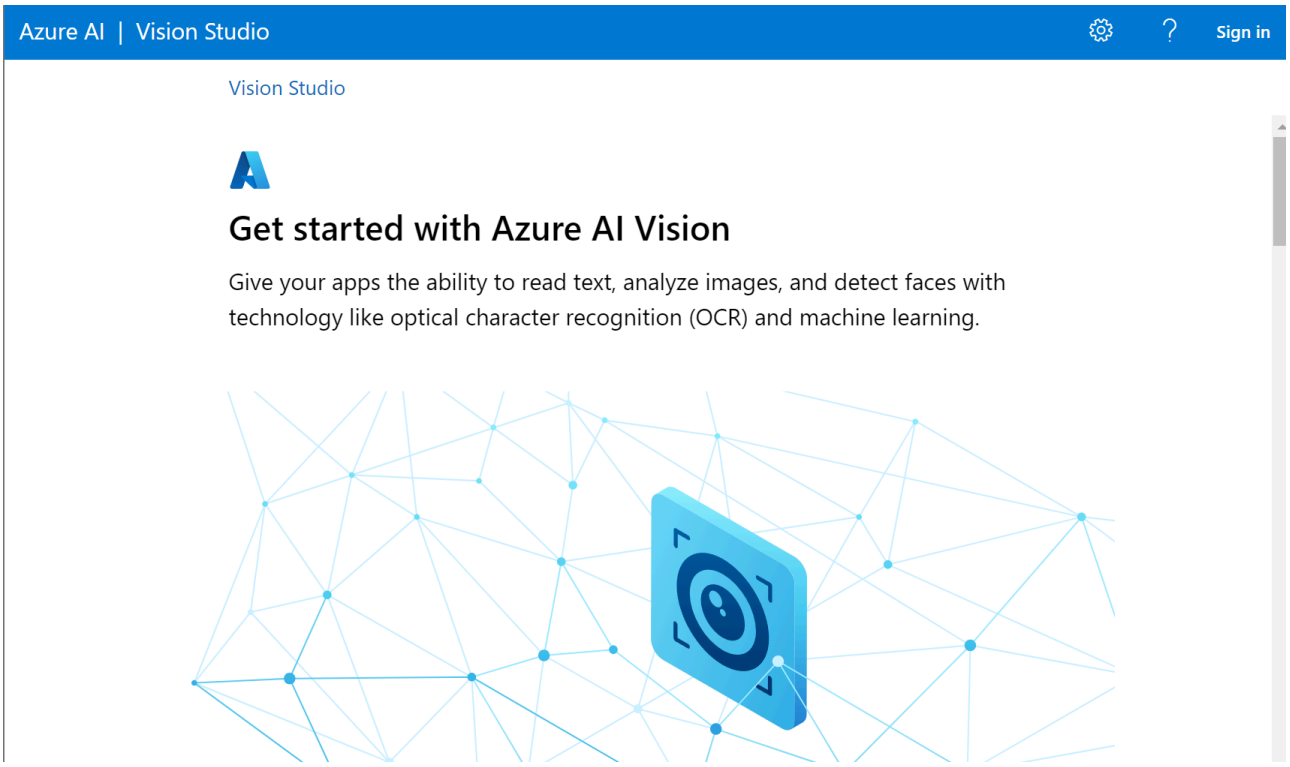
10. Navigate to `fruit`, and upload the images (and the one JSON file) in **Labfiles/02-image-classification/training-images** to that container.



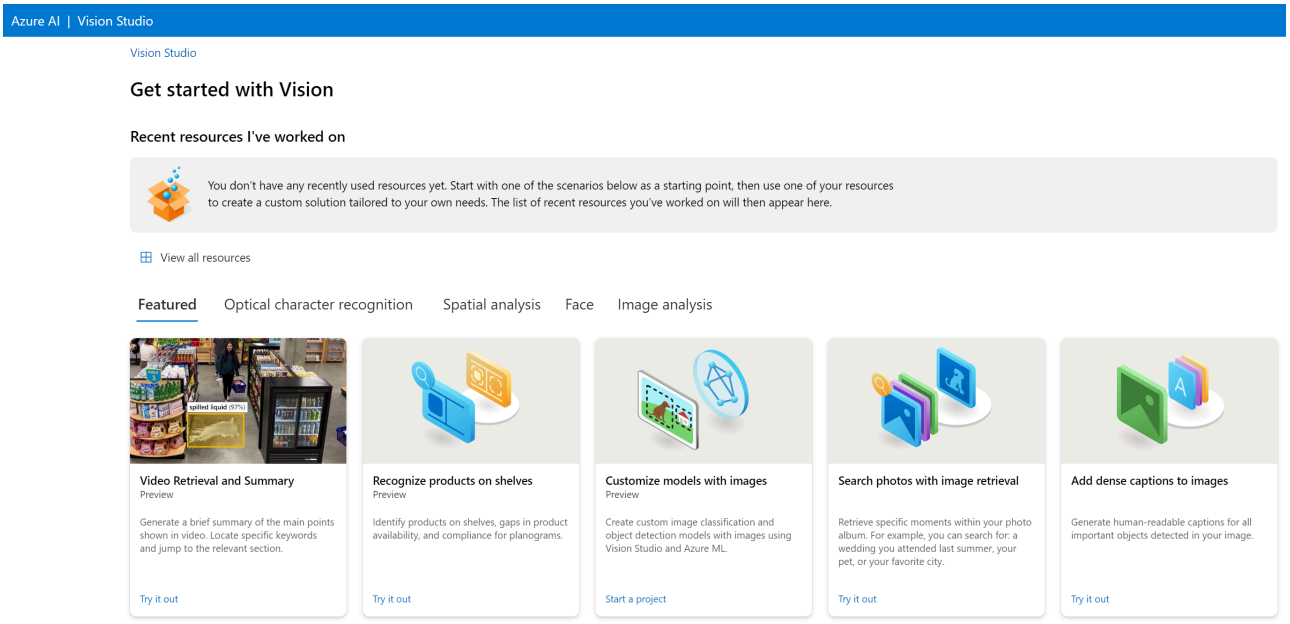
Create a custom model training project

Next, you will create a new training project for custom image classification in Vision Studio.

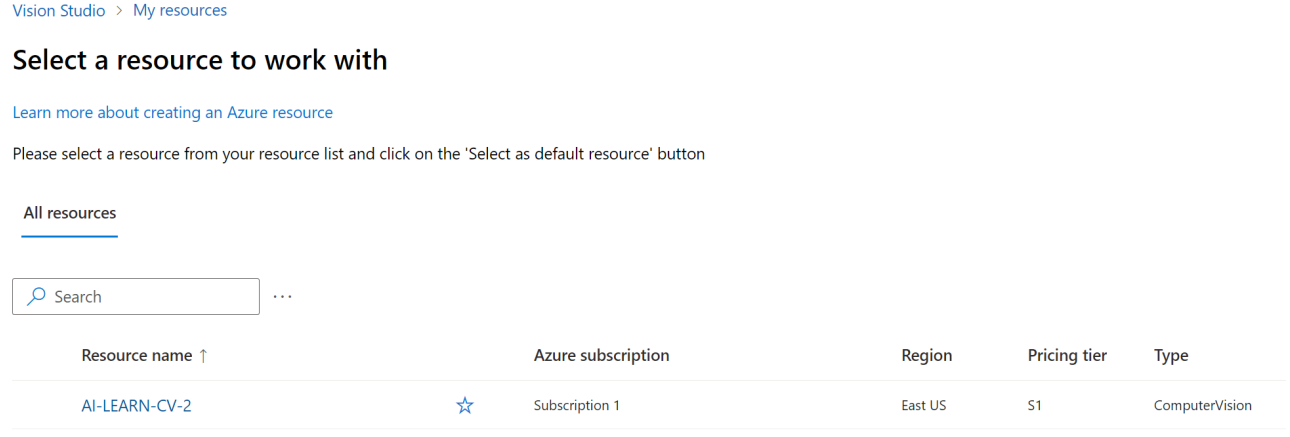
1. In the web browser, navigate to `https://portal.vision.cognitive.azure.com/` and sign in with the Microsoft account where you created your Azure AI resource.



2. Select the **Customize models with images** tile (can be found in the **Image analysis** tab if it isn't showing in your default view), and if prompted select the Azure AI resource you created.

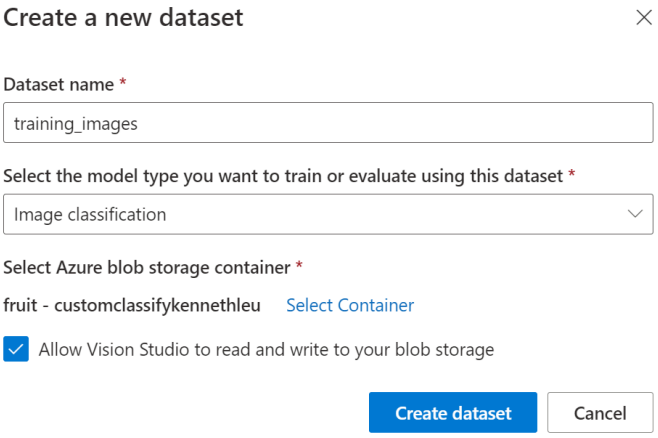


3. Select the Computer Vision resource to work with (will first require setup of single Computer Vision resource on Azure portal)



4. In your project, select **Add new dataset** on the top. Configure with the following settings:

- **Dataset name**: training_images
- **Model type**: Image classification
- **Select Azure blob storage container**: Select **Select Container**
  - **Subscription**: *Your Azure subscription*
  - **Storage account**: *The storage account you created*
  - **Blob container**: fruit
- Select the box to "Allow Vision Studio to read and write to your blob storage"



5. Select the **training_images** dataset.

At this point in project creation, you would usually select **Create Azure ML Data Labeling Project** and label your images, which generates a COCO file. You are encouraged to try this if you have time, but for the purposes of this lab we've already labeled the images for you and supplied the resulting COCO file.

1. Select **Add COCO file**
2. In the dropdown, select **Import COCO file from a Blob Container**
3. Since you have already connected your container named `fruit`, Vision Studio searches that for a COCO file. Select **training_labels.json** from the dropdown, and add the COCO file.
4. Navigate to **Custom models** on the left, and select **Train a new model**. Use the following settings:
   - **Name of model**: classifyfruit
   - **Model type**: Image classification
   - **Choose training dataset**: training_images
   - Leave the rest as default, and select **Train model**

Training can take some time - default budget is up to an hour, however for this small dataset it is usually much quicker than that. Select the **Refresh** button every couple minutes until the status of the job is *Succeeded*. Select the model.

Here you can view the performance of the training job. Review the precision and accuracy of the trained model.

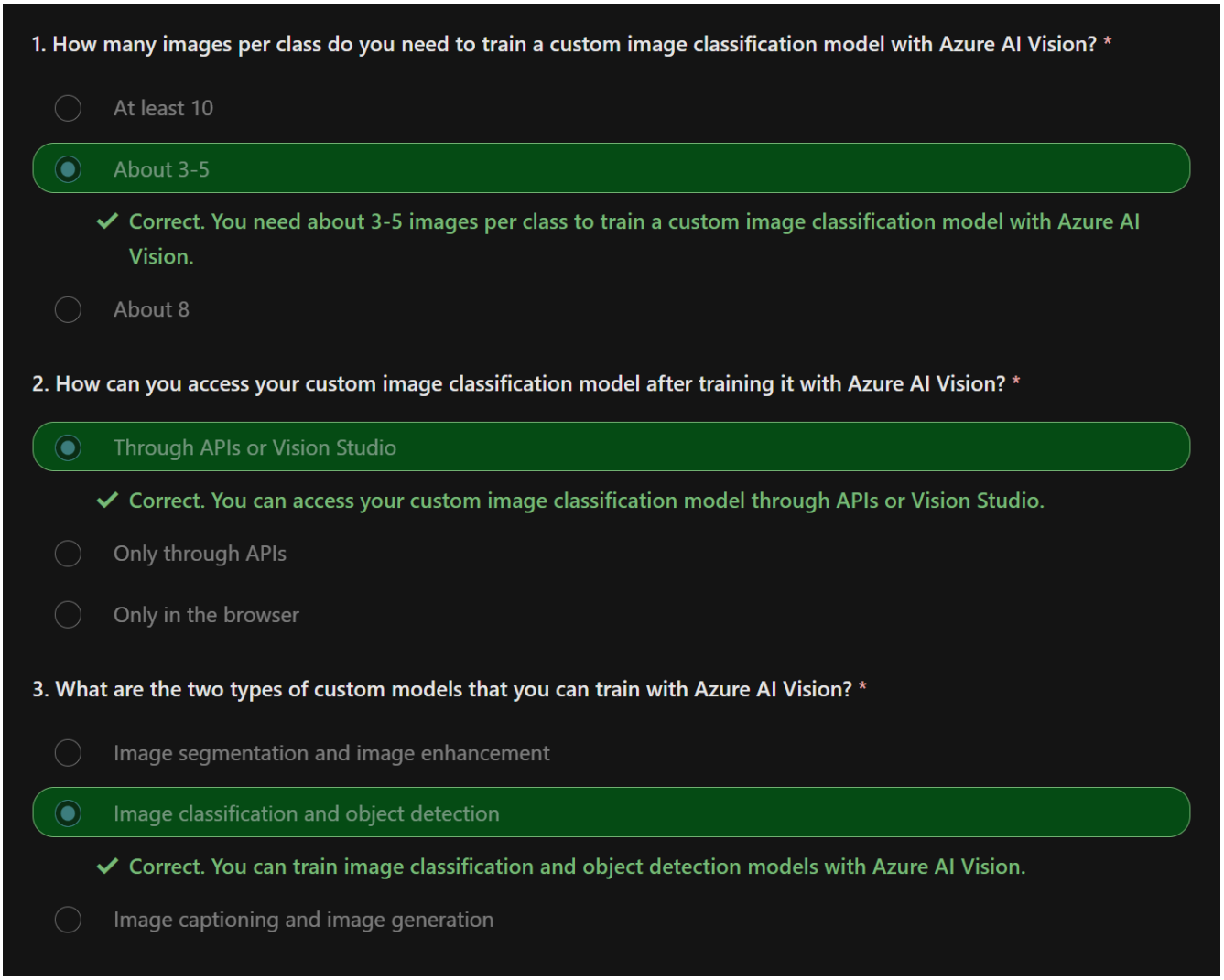
Test your custom model

Your model has been trained and is ready to test.

1. On the top of the page for your custom model, select **Try it out**.
2. Select the **classifyfruit** model from the dropdown specifying the model you want to use, and browse to the **02-image-classification\test-images** folder.

3. Select each image and view the results. Select the **JSON** tab in the results box to examine the full JSON response.

Knowledge Check



Summary

Custom models in Azure AI Vision enable you to create and train an AI model to classify images or detect objects in images that you provide and label. These custom models allow you to classify or detect exactly what your use case requires.

In this module, you learned how to:

- Create a custom model with Azure AI Vision
- Understand image classification
- Understand object detection
- Label and train an image classifier in Vision Studio

More reading:

- Create a custom Image Analysis model
- Azure AI Vision documentation

✍️ Compiled by Kenneth Leung (2025)