1. **Analyze images**

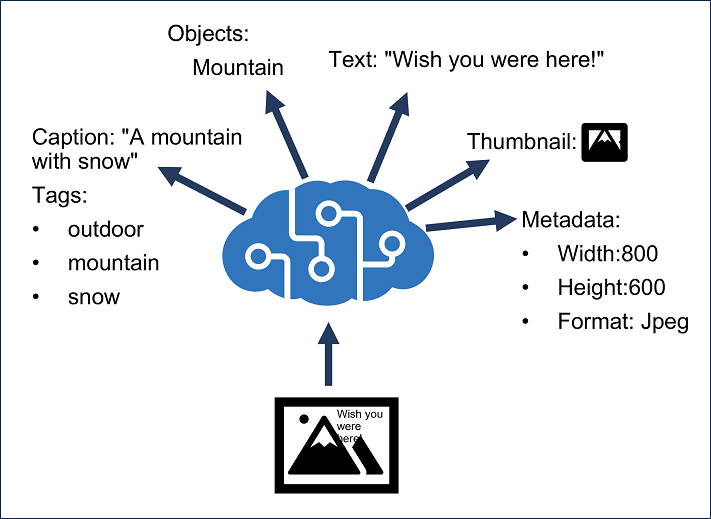
**Introduction**

Azure AI Vision is a branch of artificial intelligence (AI) in which software interprets visual input, often from images or video feeds.

**Provision an Azure AI Vision resource**

The **Azure AI Vision** service is designed to help you extract information from images. It provides functionality that you can use for:

* *Description and tag generation* - determining an appropriate caption for an image, and identifying relevant "tags" that can be used as keywords to indicate its subject.
* *Object detection* - detecting the presence and location of specific objects within the image.
* *People detection* - detecting the presence, location, and features of people in the image.
* *Image metadata, color, and type analysis* - determining the format and size of an image, its dominant color palette, and whether it contains clip art.
* *Category identification* - identifying an appropriate categorization for the image, and if it contains any known landmarks.
* *Background removal* - detecting the background in an image and output the image with the background transparent or a greyscale alpha matte image.
* *Moderation rating* - determine if the image includes any adult or violent content.
* *Optical character recognition* - reading text in the image.
* *Smart thumbnail generation* - identifying the main region of interest in the image to create a smaller "thumbnail" version.



You can provision **Azure AI Vision** as a single-service resource, or you can use the Azure AI Vision API in a multi-service **Azure AI Services** resource.

**Note**

In this module, we'll focus on the image analysis and thumbnail generation capabilities of the Azure AI Vision service. To learn how to use the Azure AI Vision service for optical character recognition, check out the [**Read Text in images and documents with the Azure AI Vision service**](https://learn.microsoft.com/en-us/training/modules/read-text-images-documents-with-computer-vision-service/) module.

**Analyze an image**

To analyze an image, you can use the Analyze Image REST method or the equivalent method in the SDK for your preferred programming language, specifying the visual features you want to include in the analysis (and if you select categories, whether or not to include details of celebrities or landmarks). This method returns a JSON document containing the requested information.

 Note

Detection of celebrities will require getting approved through a [Limited Access policy](https://aka.ms/cog-services-limited-access). You can read more about the [addition of this policy](https://azure.microsoft.com/blog/responsible-ai-investments-and-safeguards-for-facial-recognition/) to our Responsible AI standard. Celebrity recognition is seen in some screenshots, however is not included in the lab.

from azure.ai.vision.imageanalysis import ImageAnalysisClient

from azure.ai.vision.imageanalysis.models import VisualFeatures

from azure.core.credentials import AzureKeyCredential

client = ImageAnalysisClient(

endpoint=os.environ["ENDPOINT"],

credential=AzureKeyCredential(os.environ["KEY"])

)

result = client.analyze(

image\_url="<url>",

visual\_features=[VisualFeatures.CAPTION, VisualFeatures.READ],

gender\_neutral\_caption=True,

language="en",)

Available visual features are contained in the VisualFeatures enum:

* VisualFeatures.TAGS: Identifies tags about the image, including objects, scenery, setting, and actions
* VisualFeatures.OBJECTS: Returns the bounding box for each detected object
* VisualFeatures.CAPTION: Generates a caption of the image in natural language
* VisualFeatures.DENSE\_CAPTIONS: Generates more detailed captions for the objects detected
* VisualFeatures.PEOPLE: Returns the bounding box for detected people
* VisualFeatures.SMART\_CROPS: Returns the bounding box of the specified aspect ratio for the area of interest
* VisualFeatures.READ: Extracts readable text

Specifying the visual features you want analyzed in the image determines what information the response will contain. Most responses will contain a bounding box (if a location in the image is reasonable) or a confidence score (for features such as tags or captions).

The JSON response for image analysis looks similar to this example, depending on your requested features:

{

"apim-request-id": "abcde-1234-5678-9012-f1g2h3i4j5k6",

"modelVersion": "<version>",

"denseCaptionsResult": {

"values": [

{

"text": "a house in the woods",

"confidence": 0.7055229544639587,

"boundingBox": {

"x": 0,

"y": 0,

"w": 640,

"h": 640

}

},

{

"text": "a trailer with a door and windows",

"confidence": 0.6675070524215698,

"boundingBox": {

"x": 214,

"y": 434,

"w": 154,

"h": 108

}

}

]

},

"metadata": {

"width": 640,

"height": 640

}

}

**Generate a smart-cropped thumbnail and remove background**

Thumbnails are often used to provide smaller versions of images in applications and websites. For example, a tourism site might display a list of tourist attractions in a city with a small, representative thumbnail image for each attraction; and only display the full image when the user selects the "details" page for an individual attraction.

The Azure AI Vision service enables you to create a thumbnail with different dimensions (and aspect ratio) from the source image, and optionally to use image analysis to determine the *region of interest* in the image (its main subject) and make that the focus of the thumbnail. This ability to determine the region of interest is especially useful when cropping the image to change its aspect ratio.



You can specify the aspect ratio of the cropped image (width / height), ranging from 0.75 to 1.80.

**Remove image background**

The background removal feature can split the image into the subject in the foreground, and everything else that is considered background. Azure AI Vision achieves this feature by creating an *alpha matte* of the foreground subject, which is then used to return either the foreground or the background.

For example, take this image original of a skateboarder.



With the background removed, we get just the skateboarder on a transparent background.



When creating an alpha matte of an image, the result is the foreground in all white, with a black background.



Alpha matte images are helpful when client applications intend to do further processing of an image that requires separation of foreground and background objects.

**Knowledge Check**

**1. You want to use the Azure AI Vision Analyze Image function to generate an appropriate caption for an image. Which visual feature should you specify?**

Tags

DenseCaptions

**That's correct. To generate a caption, include the DenseCaptions visual feature in your analysis.**

Objects

**2. What is the purpose of the Azure AI Vision service?**

To provide functionality for audio transcription

To extract information from images

**The Azure AI Vision service is designed to help you extract information from images through various functionalities.**

To detect the presence and location of specific sounds within an audio file

**Classify images**

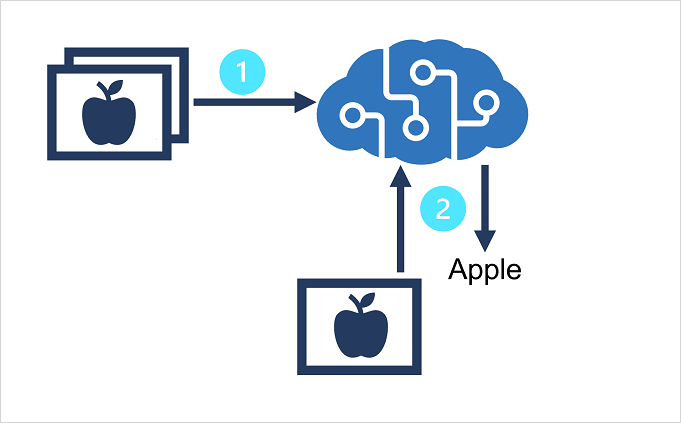
**Introduction**

*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 will learn how the Azure AI Custom Vision service enables you to build your own computer vision models for image classification.

**Provision Azure resources for Azure AI Custom Vision**

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

Creating an Azure AI Custom Vision solution involves two tasks:



1. Use existing (labeled) images to train an Azure AI Custom Vision model.
2. Create a client application that submits new images to your model to generate predictions.

To use the Azure AI Custom Vision service, you must provision two kinds of Azure resource:

* A *training* resource used to train your models. This can be:
  + An **Azure AI services multi-service** resource.
  + An **Azure AI Custom Vision (Training)** resource.
* A *prediction* resource, used by client applications to get predictions from your model. This can be:
  + An **Azure AI services multi-service** resource.
  + An **Azure AI Custom Vision (Prediction)** resource.

You can use a **Azure AI services multi-service** resource for both training and prediction, and you can mix-and-match resource types (for example, using an **Azure AI Custom Vision (Training)** resource to train a model that you then publish using an **Azure AI services multi-service** resource). If using a multi-service resource, they key and endpoint for both training and prediction will be the same.

**Understand image classification**

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.

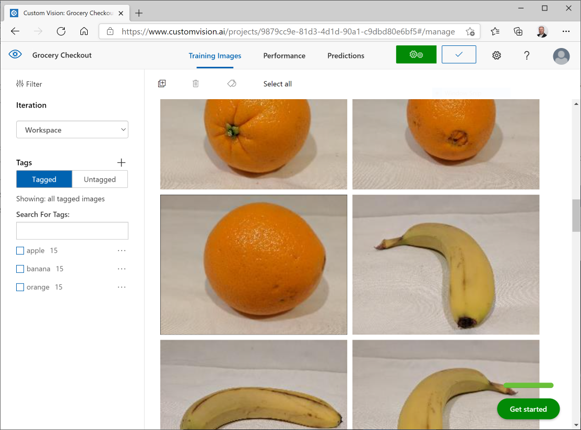


Models can be trained for multiclass classification (in other words, there are multiple classes, but each image can belong to only one class) or multilabel classification (in other words, an image might be associated with multiple labels).

**Train an image classifier**

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.

**Knowledge Check**

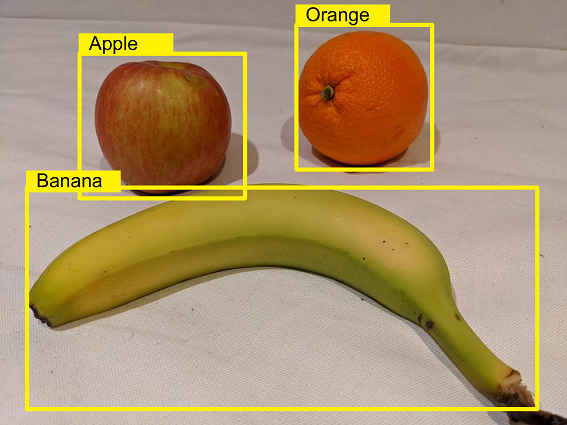
**Detect objects in images**

**Introduction**

*Object detection* is a common computer vision problem that requires software to identify the location of specific classes of object in an image. In this module, you will learn how to use the Azure AI Custom Vision service to create object detection models.

**Understand 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 by a customer.



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 one apple and two oranges.
* The location of each object within the image, indicated as coordinates of a *bounding box* that encloses the object.

**Use the Azure AI Custom Vision service for object detection**

You can use the **Azure AI Custom Vision** service to train an object detection model. To use the Azure AI Custom Vision service, you must provision two kinds of Azure resource:

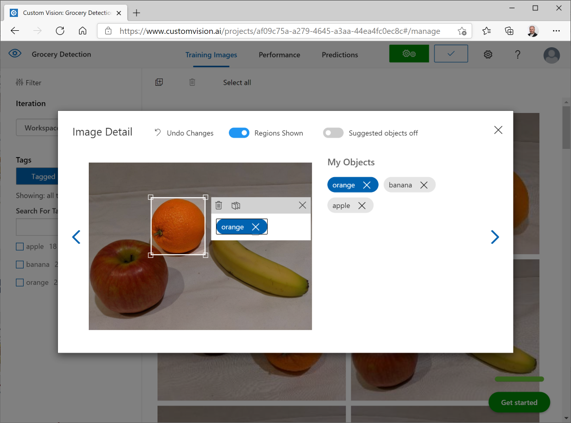
* A *training* resource used to train your models. This can be:
  + An **Azure AI services multi-service** resource.
  + An **Azure AI Custom Vision (Training)** resource.
* A *prediction* resource, used by client applications to get predictions from your model. This can be:
  + An **Azure AI services multi-service** resource.
  + An **Azure AI Custom Vision (Prediction)** resource.

You can use a **Azure AI services multi-service** resource for both training and prediction, and you can mix-and-match resource types (for example, using an **Azure AI Custom Vision (Training)** resource to train a model that you then publish using an **Azure AI services multi-service** resource). If using a multi-service resource, the key and endpoint for both training and prediction will be the same.

**Train an object detector**

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 SDK to write code that performs the training tasks.

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



**Consider options for labeling 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.

Alternatively, you can use a labeling tool, such as the one provided in [Azure Machine Learning Studio](https://learn.microsoft.com/en-us/azure/machine-learning/how-to-label-data) or the [Microsoft Visual Object Tagging Tool (VOTT)](https://github.com/microsoft/VoTT/blob/master/README.md), to take advantage of other features, such as assigning image labeling tasks to multiple team members.

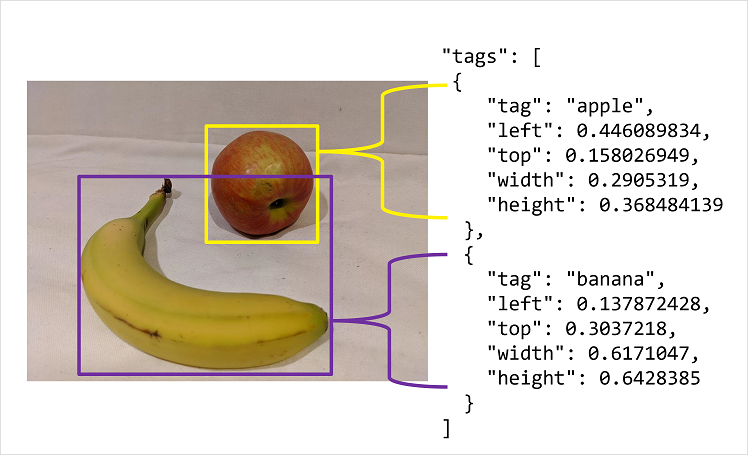
**Bounding box measurement units**

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.



**Knowledge Check**

**Detect, analyze, and recognize faces**

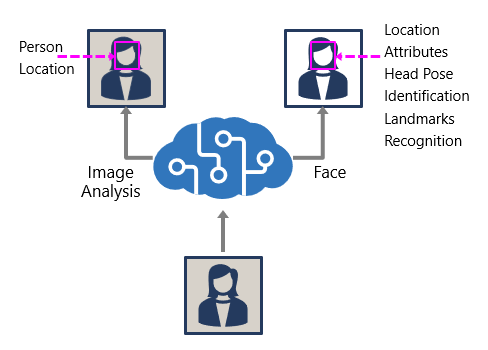
**Introduction**

Face detection, analysis, and recognition are all common computer vision challenges for AI systems. The ability to detect when a person is present, identify a person's facial location, or recognize an individual based on their facial features is a key way in which AI systems can exhibit human-like behavior and build empathy with users.

In this module, you'll learn how to detect, analyze, and recognize faces using Azure AI Services.

**Identify options for face detection analysis and identification**

There are two Azure AI services that you can use to build solutions that detect faces or people in images.



**The Azure AI Vision service**

The Azure AI Vision service enables you to detect people in an image, as well as returning a bounding box for its location.

**The Face service**

The Face service offers more comprehensive facial analysis capabilities than the Azure AI Vision service, including:

* Face detection (with bounding box).
* Comprehensive facial feature analysis (including head pose, presence of spectacles, blur, facial landmarks, occlusion and others).
* Face comparison and verification.
* Facial recognition.

 Important

Usage of facial recognition, identification, comparison, and verification will require getting approved through a [Limited Access policy](https://aka.ms/cog-services-limited-access). You can read more about the [addition of this policy](https://azure.microsoft.com/blog/responsible-ai-investments-and-safeguards-for-facial-recognition/) to our Responsible AI standard. Facial recognition will be touched on in the rest of this module, but will be unavailable without applying for Limited Access.

**Understand considerations for face analysis**

While all applications of artificial intelligence require considerations for responsible and ethical use, system that rely on facial data can be particularly problematic.

When building a solution that uses facial data, considerations include (but aren't limited to):

* **Data privacy and security**. Facial data is personally identifiable, and should be considered sensitive and private. You should ensure that you have implemented adequate protection for facial data used for model training and inferencing.
* **Transparency**. Ensure that users are informed about how their facial data is used, and who will have access to it.
* **Fairness and inclusiveness**. Ensure that your face-based system can't be used in a manner that is prejudicial to individuals based on their appearance, or to unfairly target individuals.

**Detect faces with the Azure AI Vision service**

To detect and analyze faces with the Azure AI Vision service, call the **Analyze Image** function (SDK or equivalent REST method), specifying **People** as one of the visual features to be returned.

In images that contain one or more people, the response includes details of their location in the image and the attributes of the detected person, like this:

{

"modelVersion": "2023-10-01",

"metadata": {

"width": 400,

"height": 600

},

"peopleResult": {

"values": [

{

"boundingBox": {

"x": 0,

"y": 56,

"w": 101,

"h": 189

},

"confidence": 0.9474349617958069

},

{

"boundingBox": {

"x": 402,

"y": 96,

"w": 124,

"h": 156

},

"confidence": 0.9310565276194865

},

...

]

}

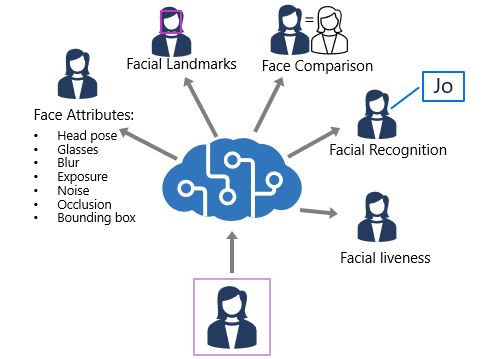
}

 Note

Azure AI Vision previously included age and gender prediction, however that has been removed as a safeguard for responsible use.

**Understand capabilities of the face service**

The **Face** service provides comprehensive facial detection, analysis, and recognition capabilities.



The Face service provides functionality that you can use for:

* *Face detection* - for each detected face, the results include an ID that identifies the face and the bounding box coordinates indicating its location in the image.
* *Face attribute analysis* - you can return a wide range of facial attributes, including:
  + Head pose (*pitch*, *roll*, and *yaw* orientation in 3D space)
  + Glasses (*NoGlasses*, *ReadingGlasses*, *Sunglasses*, or *Swimming Goggles*)
  + Blur (*low*, *medium*, or *high*)
  + Exposure (*underExposure*, *goodExposure*, or *overExposure*)
  + Noise (visual noise in the image)
  + Occlusion (objects obscuring the face)
  + Accessories (glasses, headwear, mask)
  + QualityForRecognition (*low*, *medium*, or *high*)
* *Facial landmark location* - coordinates for key landmarks in relation to facial features (for example, eye corners, pupils, tip of nose, and so on)
* *Face comparison* - you can compare faces across multiple images for similarity (to find individuals with similar facial features) and verification (to determine that a face in one image is the same person as a face in another image)
* *Facial recognition* - you can train a model with a collection of faces belonging to specific individuals, and use the model to identify those people in new images.
* *Facial liveness* - liveness can be used to determine if the input video is a real stream or a fake to prevent bad intentioned individuals from spoofing the recognition system.

You can provision **Face** as a single-service resource, or you can use the Face API in a multi-service **Azure AI Services** resource.

If you want to use the identification, recognition, and verification features of **Face**, you'll need to apply for the [Limited Access policy](https://aka.ms/cog-services-limited-access) and get approval before these features are available.

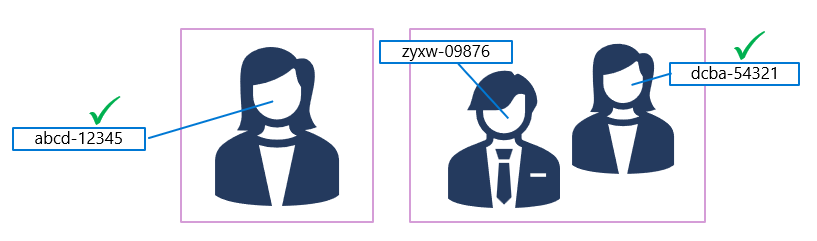
**Compare and match detected faces**

When a face is detected by the Face service, a unique ID is assigned to it and retained in the service resource for 24 hours. The ID is a GUID, with no indication of the individual's identity other than their facial features.

**Important**

Usage of facial recognition, comparison, and verification will require getting approved through a [**Limited Access policy**](https://aka.ms/cog-services-limited-access). You can read more about the [**addition of this policy**](https://azure.microsoft.com/blog/responsible-ai-investments-and-safeguards-for-facial-recognition/) to our Responsible AI standard. Facial recognition will be unavailable to new customers until they are granted the Limited Access policy.

While the detected face ID is cached, subsequent images can be used to compare the new faces to the cached identity and determine if they are *similar* (in other words, they share similar facial features) or to *verify* that the same person appears in two images.



This ability to compare faces anonymously can be useful in systems where it's important to confirm that the same person is present on two occasions, without the need to know the actual identity of the person. For example, by taking images of people as they enter and leave a secured space to verify that everyone who entered leaves.

**Implement facial recognition**

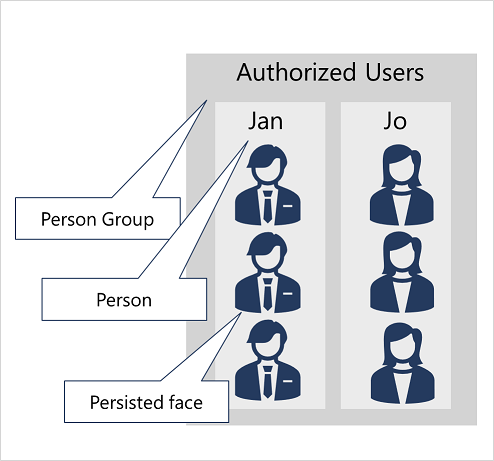
For scenarios where you need to positively identify individuals, you can train a facial recognition model using face images.

**Note**

As mentioned in the previous unit, recognition models will require getting approved through a [**Limited Access policy**](https://aka.ms/cog-services-limited-access).

To train a facial recognition model with the Face service:

1. Create a **Person Group** that defines the set of individuals you want to identify (for example, *employees*).
2. Add a **Person** to the **Person Group** for each individual you want to identify.
3. Add detected faces from multiple images to each **person**, preferably in various poses. The IDs of these faces will no longer expire after 24 hours (so they're now referred to as *persisted* faces).
4. Train the model.



The trained model is stored in your Face (or Azure AI Services) resource, and can be used by client applications to:

* *Identify* individuals in images.
* *Verify* the identity of a detected face.
* Analyze new images to find faces that are *similar* to a known, persisted face.