**4.1 Analyze images**

Computer Vision is a branch of artificial intelligence (AI) in which software interprets visual input, often from images or video feeds. In Microsoft Azure, you can use the Azure AI Vision service to implement multiple computer vision scenarios, including:

Image analysis

Optical character recognition (OCR)

Face detection and analysis

Video analysis

**Provision an Azure AI Vision resource**

To use Azure AI Vision image analysis services, you need to provision an Azure AI Vision resource in your Azure subscription. You can choose from multiple provisioning options:

1. Create an **Azure AI Foundry project** and **an associated hub**. By default, an Azure AI Foundry hub includes an **Azure AI services** multi-service resource, which includes Azure AI Vision. Azure AI Foundry projects are recommended for development of AI solutions on Azure that combine generative AI, agents, and pre-built Azure AI services, or which involve collaborative development by a team of software engineers and service operators.
2. If you don't need all of the functionality in an Azure AI Foundry hub, you can create an **Azure AI services** multi-service resource in your Azure subscription. You can then use this resource to access Azure AI Vision services and other AI services through a single endpoint and key.
3. If you only need to use Azure AI Vision functionality, or you're just experimenting with the service, you can create a standalone **Computer Vision** resource in your Azure subscription. One benefit of this approach is that the **standalone service provides a free tier** that you can use to explore the service at no cost.

**Connecting to your resource**

You can use the Azure AI Vision REST API or a language-specific SDK to connect to it from a client application.

Every Azure AI Vision resource provides an endpoint to which client applications must connect. You can find the endpoint for your resource in the Azure portal, or if you're working in an Azure AI Foundry project, in the Azure AI Foundry portal. The endpoint is in the form of a URL, and typically looks something like this:

https://<resource\_name>.cognitiveservices.azure.com/

To connect to the endpoint, client applications must be authenticated. Options for authentication include:

* Key-based authentication: Client applications are authenticated by passing an authorization key (which you can find and manage in the portal).
* Microsoft Entra ID authentication: Client applications are authenticated by using a Microsoft Entra ID token for credentials that have permission to access the Azure AI Vision resource in Azure.

When using an Azure AI services resource in an Azure AI Foundry project, you can use the Azure AI Foundry SDK to connect to the project using Microsoft Entra ID authentication, and then retrieve the connection information for your Azure AI services resource, including the authorization key, from the project.

**Analyze an image**

After connecting to your Azure AI Vision resource endpoint, your client application can use the service to perform image analysis tasks.

Note the following requirements for image analysis:

* The image must be presented in JPEG, PNG, GIF, or BMP format.
* The file size of the image must be less than 4 megabytes (MB).
* The dimensions of the image must be greater than 50 x 50 pixels.

**Submitting an image for analysis**

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.

Available visual features are contained in the **azure.ai.vision.imageanalysis.models.VisualFeatures** enumeration:

* **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).

**4.2 - Read text in images (OCR)**

**Explore Azure AI options for reading text**

There are multiple Azure AI services that read text from documents and images, each optimized for results depending on the input and the specific requirements of your application.

**Azure AI Vision** includes an **image analysis** capability that supports **optical character recognition (OCR)**. Consider using Azure AI Vision in the following scenarios:

* **Text location and extraction from scanned documents**: Azure AI Vision is a great solution for general, unstructured documents that have been scanned as images. For example, reading text in labels, menus, or business cards.
* **Finding and reading text in photographs**: Examples include photo's that include street signs and store names.
* **Digital asset management (DAM)**: Azure AI Vision includes functionality for analyzing images beyond extracting text; including object detection, describing or categorizing an image, generating smart-cropped thumbnails and more. These capabilities make it a useful service when you need to catalog, index, or analyze large volumes of digital image-based content.

**Azure AI Document Intelligence** is a service that you can use to extract information from complex digital documents. Azure AI Document Intelligence is designed for extracting text, key-value pairs, tables, and structures from documents automatically and accurately. Key considerations for choosing Azure AI Document Intelligence include:

* **Form processing**: Azure AI Document Intelligence is specifically designed to extract data from forms, invoices, receipts, and other structured documents.
* **Prebuilt models**: Azure AI Document Intelligence provides prebuilt models for common document types to reduce complexity and integrate into workflows or applications.
* **Custom models**: Creating custom models tailored to your specific documents, makes Azure AI Document Intelligence a flexible solution that can be used in many business scenarios.

**Azure AI Content Understanding** is a service that you can use to analyze and extract information from multiple kinds of content; including documents, images, audio streams, and video. It is suitable for:

* **Multimodal content extraction**: Extracting content and structured fields from documents, forms, audio, video, and images.
* **Custom content analysis scenarios**: Support for customizable analyzers enables you to extract specific content or fields tailored to business needs.

**Read text with Azure AI Vision Image Analysis**

To use Azure AI Vision for image analysis, including optical character recognition, you must provision an Azure AI Vision resource in an Azure subscription. The resource can be:

* An **Azure AI Services** multi-service resource (either deployed as part of an Azure AI Foundry hub and project, or as a standalone resource).
* A **Computer Vision resource**.

To use the Azure AI Vision Python SDK to extract text from an image, install the **azure-ai-vision-imageanalysis** package.

Connect an **ImageAnalysisClient** object to an Azure AI Vision resource. To find and read text in an image, call the **analyze** (or **analyze\_from\_url**) method, specifying the **VisualFeatures.READ** enumeration.

**4.3 - Detect, analyze, and recognize faces**

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

**Access to the full capabilities of the Face API is restricted in accordance with Microsoft's responsible AI policies.Access to the full capabilities of the Face API is restricted in accordance with Microsoft's responsible AI policies.**

**Plan a face detection, analysis, or recognition solution**

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 (No glasses, Reading glasses, Sunglasses, or Swimming Goggles)
  + Mask (the presence of a face mask)
  + Blur (low, medium, or high)
  + Exposure (under exposure, good exposure, or over exposure)
  + 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 a facial recognition system.

**Face detection and recognition models**

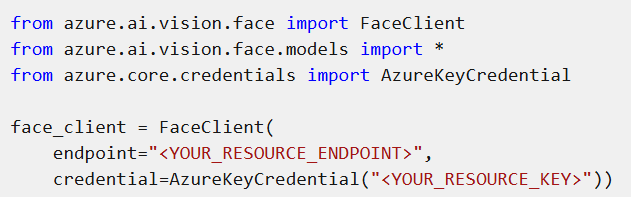
The Azure AI Vision Face API is built on face detection and recognition models that have been pre-trained. Multiple versions of these models are available, each with specific strengths and capabilities. For example, newer models exhibit greater accuracy when working with small images; but may not provide the same breadth of facial analysis capabilities. When you use the service in an application, you must select the model you want to use based on your requirements.

**Detect and analyze faces**

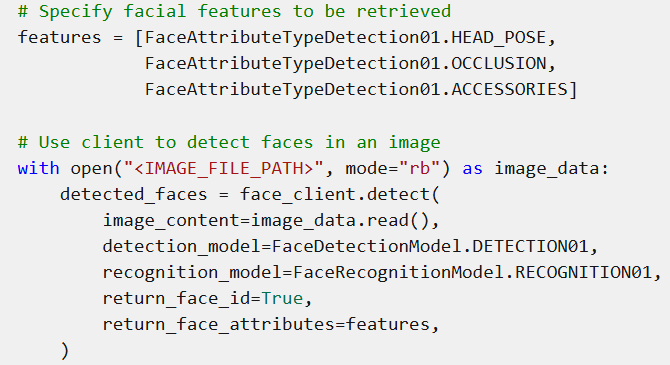
To use the Azure AI Vision Face API, you can provision **Face** as a single-service resource, or you can use the Face API in a multi-service Azure AI Services resource; which can be provisioned as a standalone resource or as part of an Azure AI Foundry hub.

To use the Azure AI Vision Python SDK to extract text from an image, install the **azure-ai-vision-face** package.

Connect an **FaceClient** object to an Azure AI Vision resource.



To detect and analyze faces in an image, you must **specify the model-specific features** you want the service to return, and then use the client to call the **Detect** method.



The response from the service depends on:

The model-specific features requested.

The number of faces detected in the image.

**Verify and identify faces**

In addition to detecting and analyzing faces, you can use the Azure AI Vision Face service to compare and recognize faces.

**Verifying 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.
* 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're 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.

**Identifying faces**

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

To train a facial recognition model with the Face service:

* **Create a Person Group** that defines the set of individuals you want to identify (for example, employees).
* **Add a Person** to the Person Group for each individual you want to identify.
* **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**).
* **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.

**Responsible AI considerations for face-based solutions**

While all applications of artificial intelligence require considerations for responsible, system that rely on facial or other biometric 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.

**4.4 - Classify images**

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.

**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.

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.

Azure AI Custom Vision (**https://www.customvision.ai**) provides a web-based portal, in which you can train, publish, and test custom vision models.

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

Python developers can perform both training and prediction tasks by using the **azure-cognitiveservices-vision-customvision** package.

**Train an image classification model**

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.

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.

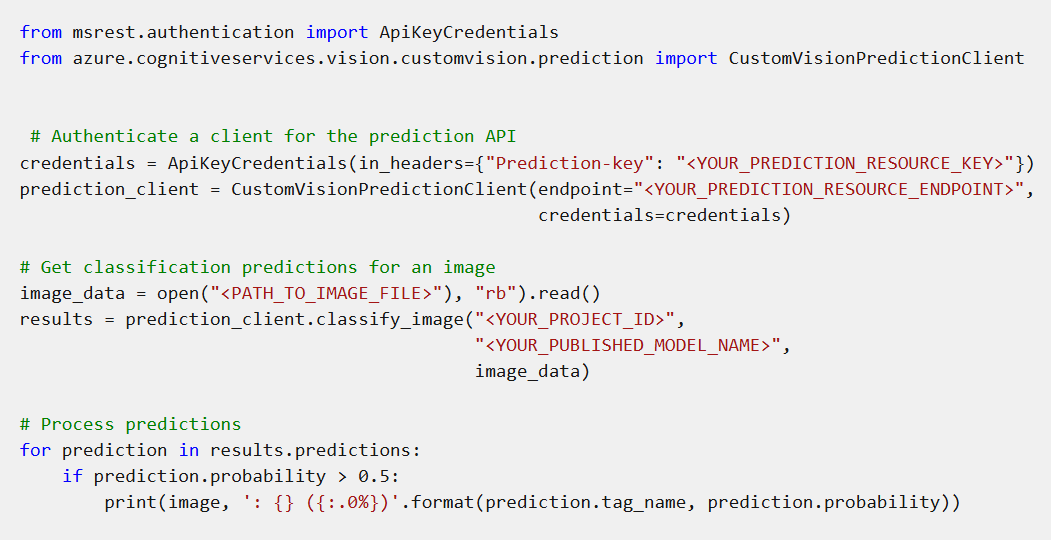
The portal provides a graphical interface that you can use to:

* Create an image classification project for your model and associate it with a training resource.
* Upload images, assigning class label tags to them.
* Review and edit tagged images.
* Train and evaluate a classification model.
* Test a trained model.
* 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.

**Create an image classification client application**

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.



* 1. **- Detect objects in images**

**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. You can use Azure AI Custom Vision (**https://customvision.ai/**) to train a model to detect specific classes of object in images.

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.

**Develop an object detection client application**

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.



**4.6 - Analyze video (Azure Video Indexer)**

**Understand Azure Video Indexer capabilities**

The Azure Video Indexer (**https://www.videoindexer.ai/**) service is designed to help you extract information from videos. It provides functionality that you can use for:

* **Facial recognition** - detecting the presence of individual people in the image. This requires Limited Access approval.
* **Optical character recognition** - reading text in the video.
* **Speech transcription** - creating a text transcript of spoken dialog in the video.
* **Topics** - identification of key topics discussed in the video.
* **Sentiment** - analysis of how positive or negative segments within the video are.
* **Labels** - label tags that identify key objects or themes throughout the video.
* **Content moderation** - detection of adult or violent themes in the video.
* **Scene segmentation** - a breakdown of the video into its constituent scenes.

The Video Analyzer service provides a portal website that you can use to upload, view, and analyze videos interactively.

**Extract custom insights**

Azure Video Indexer includes predefined models that can recognize well-known celebrities, do OCR, and transcribe spoken phrases into text. You can extend the recognition capabilities of Video Analyzer by creating custom models for:

* **People**. Add images of the faces of people you want to recognize in videos, and train a model. Video Indexer will then recognize these people in all of your videos.
* **Language**. If your organization uses specific terminology that may not be in common usage, you can train a custom model to detect and transcribe it.
* **Brands**. You can train a model to recognize specific names as brands, for example to identify products, projects, or companies that are relevant to your business.

**Use Video Analyzer widgets and APIs**

While you can perform all video analysis tasks in the Azure Video Indexer portal, you may want to incorporate the service into custom applications. There are two ways you can accomplish this.

* **Azure Video Indexer widgets**

The widgets used in the Azure Video Indexer portal to play, analyze, and edit videos can be embedded in your own custom HTML interfaces. You can use this technique to share insights from specific videos with others without giving them full access to your account in the Azure Video Indexer portal.

* **Azure Video Indexer API**

Azure Video Indexer provides a REST API that you can use to obtain information about your account, including an access token.

**https://api.videoindexer.ai/Auth/<location>/Accounts/<accountId>/AccessToken**

You can then use your token to consume the REST API and automate video indexing tasks, creating projects, retrieving insights, and creating or deleting custom models.

**Deploy with ARM template**

Azure Resource Manager (ARM) templates are available to create the Azure AI Video Indexer resource in your subscription, based on the parameters specified in the template file.

**4.7 - Develop a vision-enabled generative AI application**

Generative AI models enable you to develop chat-based applications that reason over and respond to input. Often this input takes the form of a text-based prompt, but increasingly multimodal models that can respond to visual input are becoming available.

**Deploy a multimodal model**

To handle prompts that include images, you need to deploy a multimodal generative AI model - in other words, a model that supports not only text-based input, but image-based (and in some cases, audio-based) input as well. Multimodal models available in Azure AI Foundry include (among others):

* Microsoft Phi-4-multimodal-instruct
* OpenAI gpt-4o
* OpenAI gpt-4o-mini

**Develop a vision-based chat app**

To develop a client app that engages in vision-based chats with a multimodal model, you can use the same basic techniques used for text-based chats. You require a connection to the endpoint where the model is deployed, and you use that endpoint to submit prompts that consists of messages to the model and process the responses.

The key difference is that prompts for a vision-based chat include multi-part user messages that contain both a text (or audio where supported) content item and an image content item.

The JSON representation of a prompt that includes a multi-part user message looks something like this:



When using binary data to submit a local image file, the image\_url content takes the form of a base64 encoded value in a data URL format:



**4.7 - Generate images with AI**

**What are image-generation models?**

Azure AI Foundry supports multiple models that are capable of generating images, including (but not limited to):

DALL-E 3

GPT-Image 1

Image generation models are generative AI model that can create graphical data from natural language input. Put more simply, you can provide the model with a description and it can generate an appropriate image.

**Explore image-generation models in Azure AI Foundry portal**

* To experiment with image generation models, you can create an Azure AI Foundry project and use the Images playground in Azure AI Foundry portal to submit prompts and view the resulting generated images.
* When using the playground, you can adjust the settings to control the output. For example, when using a DALL-E model you can specify:
  + The resolution (size) of the generated images. Available sizes are 1024x1024 (which is the default value), 1792x1024, or 1024x1792.
  + The image style to be generated (such as vivid or natural).
  + The image quality (choose from standard or hd).

**Create a client application that uses an image generation model**

You can use a REST API to consume DALL-E models from applications. Alternatively, you can use a language-specific SDK (for example, the OpenAI Python SDK or the Azure OpenAI .NET SDK) to abstract the REST methods.

You initiate the image generation process by submitting a request to the service endpoint with the authorization key in the header. The request contains parameters describing the image-generation requirements. For example, parameters for a DALL-E model include:

* **prompt**: The description of the image to be generated.
* **n**: The number of images to be generated. DALL-E 3 only supports n=1.
* **size**: The resolution of the image(s) to be generated (1024x1024, 1792x1024, or 1024x1792 for DALL-E 3)
* **quality** Optional: The quality of the image (standard or hd). Defaults to standard.
* **style** Optional: The visual style of the image (natural or vivid). Defaults to vivid.