» Integrating Azure AI (Computer Vision) API

Preparation	(a) Course Notes – Module 7, 8; (b) a browser
Objectives	Create and setup a Microsoft Azure account for future use. In addition, you will learn about harnessing the power of Azure cognitive services for integration into building web applications. The design and/or presentation of some screenshots in this activity might have slight variations.

Some of the information contained in this document are compiled from Microsoft's online documentation: https://docs.microsoft.com. Some diagrams or figures may have changed from the time compiling this document. If there are any issues or differences, please contact us to let us know of any issues.

UNDERSTANDING AZURE SERVICES

The Azure platform offers a range of products and services within the cloud that facilitate the development, deployment, and administration of web applications across various computing environments. Additionally, it offers clients the opportunity to utilize a diverse array of tools and frameworks designed for the purpose of effectively managing services in the cloud. Although a comprehensive investigation or exploration of all the services provided by the Azure platform may not be feasible at this stage, as an application developer, it is highly recommended that explore other services not employed in this activity as they have the potential to provide major benefits to your project. There are several further platforms available from different vendors, including Amazon Web Services (AWS), Google Cloud Platform (GCP), IBM Watson, and others. Azure provides students with the opportunity to utilize services for learning purposes, such as a $100 coupon, without the requirement of providing personal credit-card details.

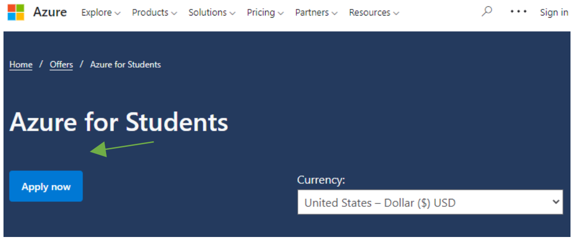
Azure web-based services are organized into distinct categories. Some services are provided at no cost, while others require a fee for each invocation. The "**All Services**" catalog offered by Azure encompasses a wide range of cloud-based service options, including but not limited to computing, storage, mobility, databases, analytics, blockchain, and networking. This collection provides support for command-line interfaces (CLIs). The subsequent table describes several prevalent categories encompassing some of the most popular Azure web-based services.

Table 1: Azure Cloud Service Categories

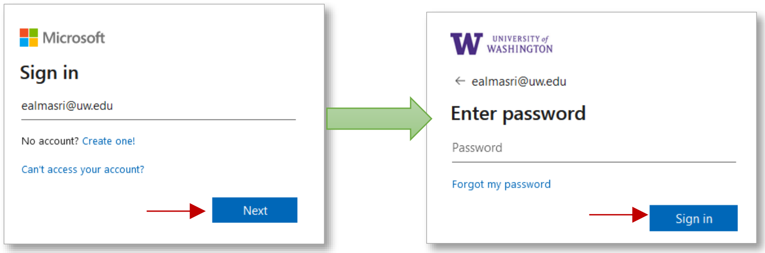
Service Category	Description
Compute	Provides services to create and deploy VMs, function Apps, Kubernetes services, container Apps, container instances, etc.
Networking	Provides services to create virtual networks, network function virtualization elements, virtual WANs, gateways, etc.
Storage	Provides services to create storage accounts, data shares, data box gateway (for edge computing), etc.
Analytics	Provides access to popular data analytic services such as HDInsight, Power BI, Data Lake Analytics, Event Hubs, etc.
AI + Machine Learning	Provides access to services such as Applied AI services, Face APIs, computer vision, custom vision, speech services, etc.
Internet of Things	Provides services to create IoT hubs, time series analytics and insights, notification hubs, device provisioning services, etc.
Identity	Provides services to Azure active directory, AD Connect Health, create AD roles, security services, Identity Governance, etc.
Databases	Provides services to Azure Cosmos DB, SQL databases, SQL managed instances, MySQL servers, SQL VMs, Azure SQL , etc.

Web services offer a mechanism for delivering application-specific functionalities to clients via the web. These services can be accessed using URL-addressable resources and are often implemented using a widely used protocol like HTTP and adhere to architectural styles like REST. To a certain degree, the utilization of a web services paradigm empowers web application developers to flexibly integrate or combine services in a loosely coupled approach. The integration of web services into a web application as a means to enhance the range of services provided to clients is increasingly recognized as a fundamental aspect of the design of web applications. In this activity, we will **first** understand the process of employing various Azure web services. Then, we will learn how to effectively incorporate these services into web applications with the goal of providing specialized functionalities to clients.

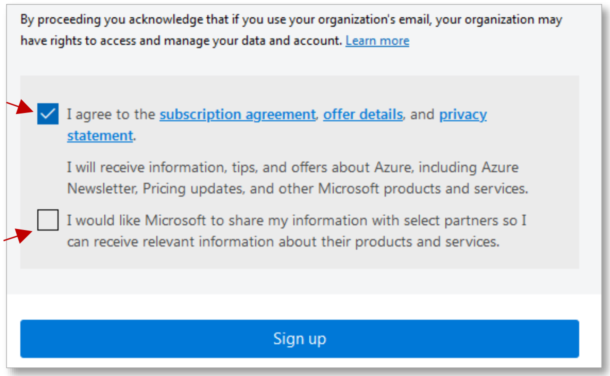
PART A » CREATING AN EDUCATION AZURE ACCOUNT



1. Open a web browser and enter the following URL: https://azure.microsoft.com/en-us/offers/ms-azr-0170p/
2. Then, click on the **Apply now** button.
3. You will be prompted to provide your credentials. Enter your **UW email address** and then click **Next**. Then, you will be prompted to enter your account password. Enter the UW email password and click **Sign in**.



4. Once you are logged in, you will be prompted to fill out a form for completing the **Microsoft Azure for Students registration**. Complete the "**Your profile**" form. Ensure to review the subscription agreement and privacy statement. You may wish to **opt out** from the second option to share your information with select partners.



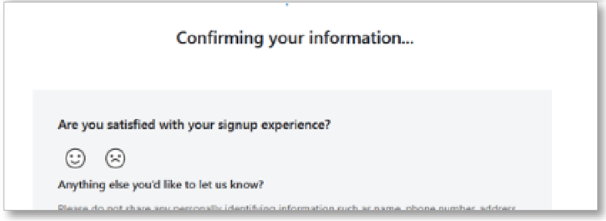
By proceeding you acknowledge that if you use your organization's email, your organization may have rights to access and manage your data and account. Learn more

☑ I agree to the subscription agreement, offer details, and privacy statement.

I will receive information, tips, and offers about Azure, including Azure Newsletter, Pricing updates, and other Microsoft products and services.

☐ I would like Microsoft to share my information with select partners so I can receive relevant information about their products and services.

5. You should receive a message confirming your account creation or registration. Within a few seconds, Azure will setup your account and it will be ready to use. Wait until the account is setup and then you will be redirected to Microsoft Azure for Education. Once completed, you should receive an email confirmation for your **$100** subscription. If you already had an Azure account, you may receive a SMS verification and then be forwarded to Azure dashboard.



Confirming your information...

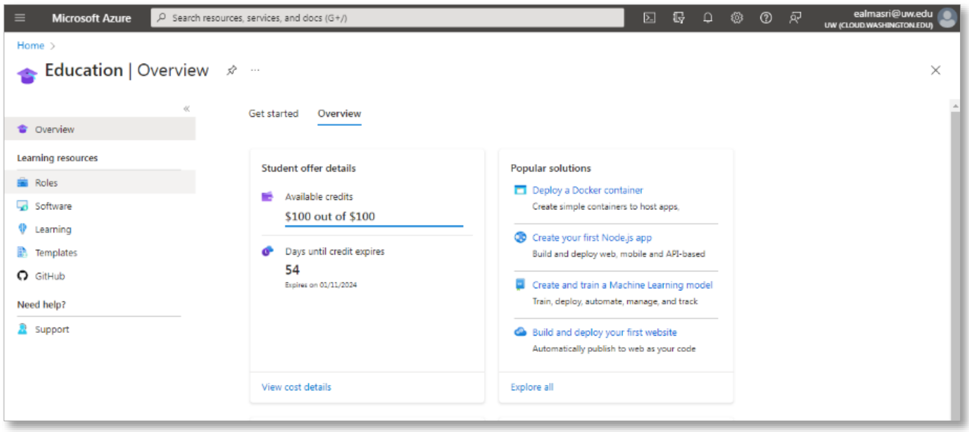
Are you satisfied with your signup experience?

☺ ☹

Anything else you'd like to let us know?

Please do not share any personally identifying information such as name, phone number, address

6. Once the registration process completes, you will be redirected to the **Education | Overview** dashboard.


Microsoft Azure
Search resources, services, and docs (G+/)
ealmasri@uw.edu
UW (CLOUD.WASHINGTON.EDU)
Home >
Education | Overview
Overview
Learning resources
Roles
Software
Learning
Templates
GitHub
Need help?
Support
Get started
Overview
View cost details
Popular solutions
Deploy a Docker container
Create simple containers to host apps.
Create your first Node.js app
Build and deploy web, mobile and API-based
Create and train a Machine Learning model
Train, deploy, automate, manage, and track
Build and deploy your first website
Automatically publish to web as your code
Explore all

Student offer details

Available credits
$100 out of $100

Days until credit expires
54
Expires on 01/11/2024

PART B » UNDERSTANDING AZURE: MICROSOFT'S CLOUD SERVICES

(I) LEARNING ABOUT AZURE AI + MACHINE LEARNING (ML) SERVICES

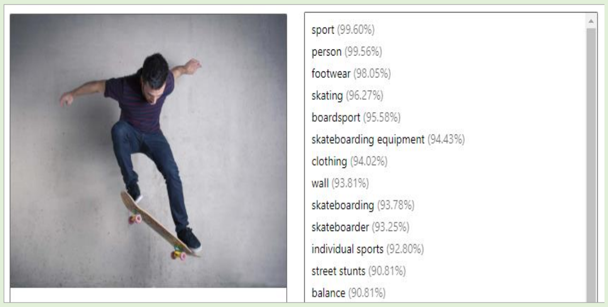
The integration of Azure **AI and machine learning (ML)** services provides developers with the opportunity to enhance applications by incorporating cognitive capabilities. Azure AI and ML services encompass a collection of artificial intelligence and machine learning techniques, which are employed to address intricate AI or ML challenges. Therefore, platforms such as Azure services streamline the process for web developers to encapsulate AI and ML components and include them into their web applications. Hence, the utilization of existing web services offered by existing web service providers, such as those provided by Microsoft Azure, can prove highly advantageous in the development of web applications. These services empower developers to seamlessly incorporate pre-existing functionalities into their own apps, thereby circumventing the arduous and time-consuming process of building such functionalities from the ground up. For instance, a web developer lacking the necessary skills or resources to construct an AI or ML algorithm for object detection in images may choose to incorporate an Azure computer vision service. This service grants access to sophisticated cognitive algorithms capable of analyzing images and furnishing pertinent data, such as object detection, object analysis, and text extraction. Table 2 displays a selection of Azure AI and ML services that are available via the Azure account.

Table 2: Catalog of Azure AI + ML Services

Service Category	Description
AI Vision APIs	Provides access to advanced cognitive algorithms for processing images and returning results or information based on what is discovered or what you might be interested in. These include: ▪ **Optical Character Recognition** (OCR) service which extracts text from images ▪ **Image Analysis service** which extract many visual features from images such as objects, faces, auto-generated textual descriptions, etc. ▪ **Spatial Analysis** which analyzes the presence and movement of people on a video feed to produce events that other systems can respond to or make further decisions. Mainly, there are three AI Vision APIs available on Azure: ▪ **Computer Vision** for processing images using pre-trained or pre-built image classifiers or models ▪ **Custom Vision** for processing images using your own image classifiers ▪ **Face** which provides access to advanced face algorithms, enabling face attribute detection and recognition (e.g., gender, age, facial hair, smile, emotion, glasses, head pose, etc.)
Speech APIs	Provides access to advanced cognitive algorithms for processing audio or speech and returning results or information based on various capabilities. In addition, the speech service adds speech-enabled features to applications which include **speech-to-text** (STT) and **text-to-speech** (TTS), among many more.
Language APIs	Provides services such as **Natural Language Processing (NLP)** for understanding and analyzing text (supports multiple languages, and a **translator** service for translating machine-based text translation in near real-time across different languages.
Decision APIs	Provides services for helping making decisions. The services include **anomaly detector** for allowing developers to monitor and detect abnormalities in time series data (e.g., sensor data), **content moderator service** for monitoring for possible offensive, undesirable and risky content and **personalizer service** for allowing developers to choose the best experience for clients or users through learning from their real-time behavior.
Search APIs	Provides search services such as **Bing News Search**, **Bing Video Search**, **Bing Web Search**, **Bing Image Search**, **Bing Spell Check**, **Bing Autosuggest**, **AI search**, among many others.

(II) WHAT CAN THE **AI VISION API** DO?

The [AI vision API](#) offers developers access to powerful algorithms designed for the purpose of imagine processing. The AI Vision API has the capability to evaluate visual content through the process of uploading an image or supplying a URL of an image that resides on the web. The analysis of content can be controlled by manipulating inputs and user settings.

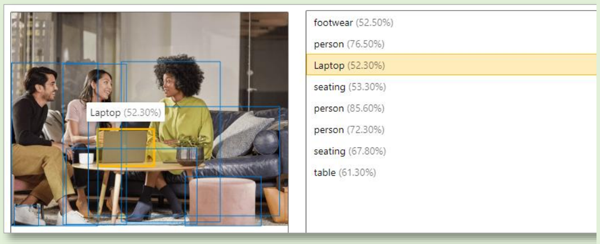


sport (99.60%)
person (99.56%)
footwear (98.05%)
skating (96.27%)
boardsport (95.58%)
skateboarding equipment (94.43%)
clothing (94.02%)
wall (93.81%)
skateboarding (93.78%)
skateboarder (93.25%)
individual sports (92.80%)
street stunts (90.81%)
balance (90.81%)

Source: https://docs.microsoft.com/en-us/azure/cognitive-services/computer-vision/overview-image-analysis

The AI Vision API offers SDKs in various programming languages, including C#, Node.js, Python, Go, and Java. This application programming interface (API) enables the analysis of images to offer valuable observations regarding the visual attributes and qualities exhibited by the image. As illustrated in the provided examples, it is possible to detect and label visual attributes within an image.

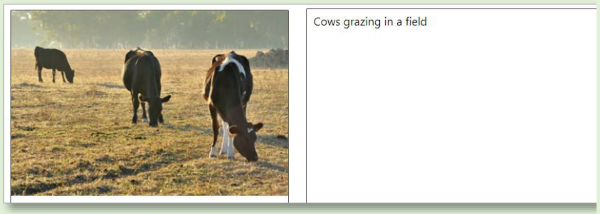
The AI Vision API can be utilized for the purpose of object detection, which is comparable to the process of tagging. However, it is important to note that the API provides the bounding box coordinates for each tag that is applied, as exemplified in the example on the right side.



footwear (52.50%)
person (76.50%)
Laptop (52.30%)
seating (53.30%)
person (85.60%)
person (72.30%)
seating (67.80%)
table (61.30%)

Source: https://docs.microsoft.com/en-us/azure/cognitive-services/computer-vision/overview-image-analysis

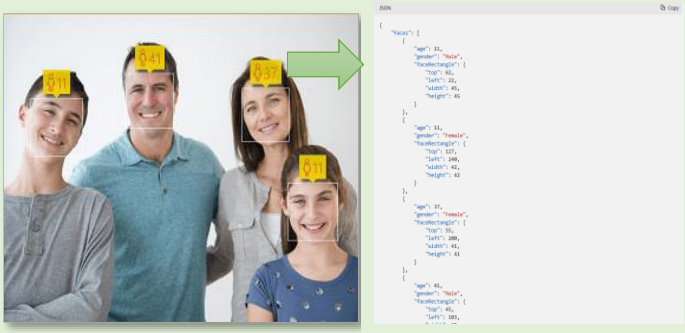
The AI Vision API has also the capability to provide a description of the content found within an image, as demonstrated in the example on the right side.



Cows grazing in a field

Source: https://docs.microsoft.com/en-us/azure/cognitive-services/computer-vision/overview-image-analysis

The AI Vision API can identify and analyze facial features within a given image, yielding specific information pertaining to the recognized faces, as illustrated in the example on the right-hand side.



```
{
    "faces": [
        {
            "age": 11,
            "gender": "Male",
            "faceRectangle": {
                "top": 62,
                "left": 22,
                "width": 45,
                "height": 45
            }
        },
        {
            "age": 11,
            "gender": "Female",
            "faceRectangle": {
                "top": 127,
                "left": 240,
                "width": 42,
                "height": 42
            }
        },
        {
            "age": 37,
            "gender": "Female",
            "faceRectangle": {
                "top": 55,
                "left": 200,
                "width": 41,
                "height": 41
            }
        },
        {
            "age": 41,
            "gender": "Male",
            "faceRectangle": {
                "top": 45,
                "left": 103,
```

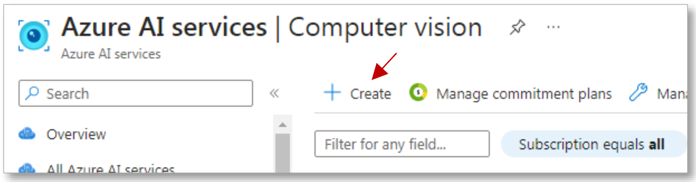
Source: https://docs.microsoft.com/en-us/azure/cognitive-services/computer-vision/concept-detecting-faces

The Azure Education account provides students with the opportunity to experiment with select functionalities offered by the Azure AI Vision API. The API leverages pre-existing AI and ML models that have undergone training to reliably identify or produce outcomes. Students have the option to develop their own AI and ML models. Nevertheless, for the purpose of this activity, our main emphasis will be on pre-existing AI and ML models. We will aim to acquire knowledge on how to successfully integrate the outcomes derived from these models into our own API, which we will develop.

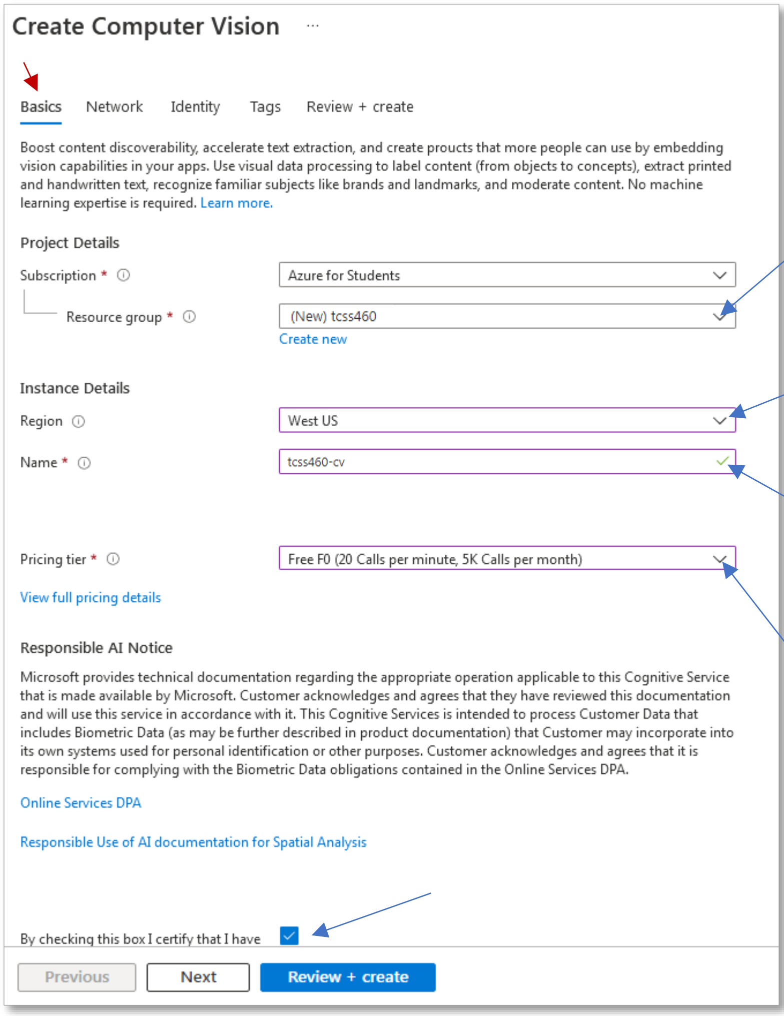
(III) CREATING AN INSTANCE AND USING THE COMPUTER VISION API

1. Go to the Azure portal https://portal.azure.com (should be logged in)
2. From the top-left corner, click on the navigation menu (≡) and then select **All Services**.
3. From the left menu, click on **AI + machine learning** (or scroll to that section on the web page)
4. Click on the **Computer vision** option or link

5. **Create** a new instance of the Computer Vision Azure AI service by clicking on the **+Create** button (top-middle)


Azure AI services | Computer vision
Azure AI services

6. Under the **Basics** tab, enter or select the following project information (mouse over the ⓘ icon to learn about each options and what it means. For the **Region**, select **West US**. Then, scroll down the page and ensure you check the **checkbox** option to certify you have ready the terms above. Then, click the '**Review + create**' button.


Create Computer Vision ···

Boost content discoverability, accelerate text extraction, and create proucts that more people can use by embedding vision capabilities in your apps. Use visual data processing to label content (from objects to concepts), extract printed and handwritten text, recognize familiar subjects like brands and landmarks, and moderate content. No machine learning expertise is required. Learn more.

Project Details

Subscription * ⓘ: Azure for Students

Resource group * ⓘ: (New) tcss460
Create new

Instance Details

Region ⓘ: West US

Name * ⓘ: tcss460-cv

Pricing tier * ⓘ: Free F0 (20 Calls per minute, 5K Calls per month)

View full pricing details

Responsible AI Notice

Microsoft provides technical documentation regarding the appropriate operation applicable to this Cognitive Service that is made available by Microsoft. Customer acknowledges and agrees that they have reviewed this documentation and will use this service in accordance with it. This Cognitive Services is intended to process Customer Data that includes Biometric Data (as may be further described in product documentation) that Customer may incorporate into its own systems used for personal identification or other purposes. Customer acknowledges and agrees that it is responsible for complying with the Biometric Data obligations contained in the Online Services DPA.

Online Services DPA

Responsible Use of AI documentation for Spatial Analysis

By checking this box I certify that I have ☑

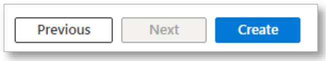
Ensure to create a new resource group (click on **Create new** and then type resource group name **tcss460**)

You may select any region. However, select one that is closest to the location where the application is to be deployed. We will use **West US**.

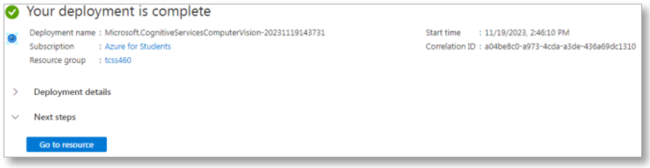
Provide a name for the instance. In this case, we wish to create a new instance name for computer vision. Hence, we will call it **tcss460-cv**.

Select **Free F0** as the pricing tier.
If you receive **errors** creating the instance or is unable to locate Free F0 option, go back and change the region to **West US 2** or **West US 3** to be able to select a Free F0 pricing tier.

7. Azure will verify the selection and once validation is passed, it will prompt you to create the computer vision instance. Click the '**Create**' button.



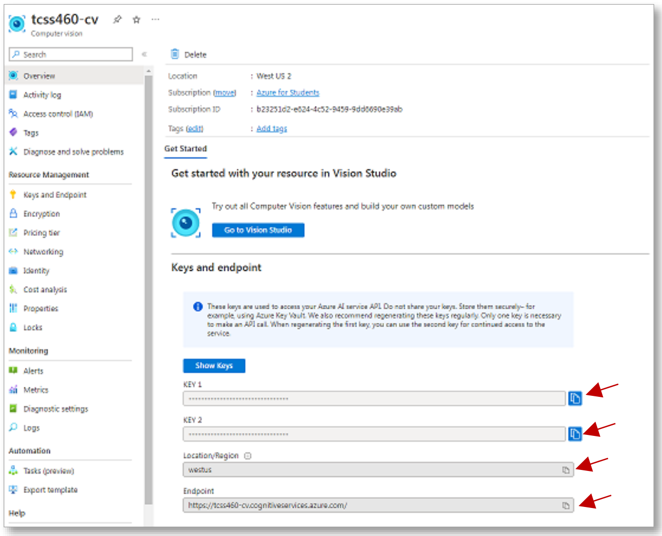
8. You will see a **Deployment is in progress** message. Once completed, you should see a message "**Your deployment is complete"**. Click on the '**Go to resource**' button.


Your deployment is complete

Deployment name : Microsoft.CognitiveServicesComputerVision-20231119143731
Subscription : Azure for Students
Resource group : tcss460
Start time : 11/19/2023, 2:46:10 PM
Correlation ID : a04be8c0-a973-4cda-a3de-436a69dc1310

9. The overview webpage for the **tcss460-cv** instance provides users with the ability to effectively oversee, control, and monitor the service in real-time. The left navigation menu provides a list of options to manage this instance. At the body of this webpage, you should view the Keys and endpoint.

   In this section, which also can be accessed by clicking on the **Keys and Endpoint** link on the left navigation menu, provides the keys that will be used to access your Azure AI service API. **Copy** and store the keys securely, including **key 1, key 2, location/region** and **service endpoint**.



Location : West US 2

Subscription (move) : Azure for Students

Subscription ID : b23251d2-e624-4c52-9459-9dd6690e39ab

Tags (edit) : Add tags

Get started with your resource in Vision Studio

Try out all Computer Vision features and build your own custom models

Keys and endpoint

These keys are used to access your Azure AI service API. Do not share your keys. Store them securely– for example, using Azure Key Vault. We also recommend regenerating these keys regularly. Only one key is necessary to make an API call. When regenerating the first key, you can use the second key for continued access to the service.

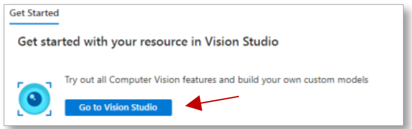
KEY 1: ••••••••••••••••••••••••••••••••

KEY 2: ••••••••••••••••••••••••••••••••

Location/Region: westus

Endpoint: https://tcss460-cv.cognitiveservices.azure.com/

10. Under the '**Get Started**' section within the **tcss460-cv** overview page, click on the '**Go to Vision Studio**' button.



Get started with your resource in Vision Studio

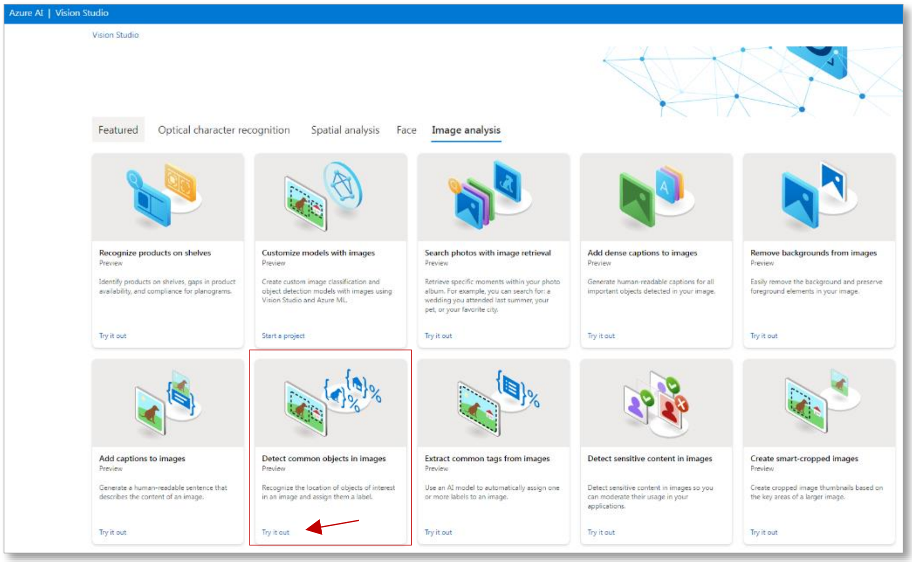
Try out all Computer Vision features and build your own custom models

11. A new window should open with a list of offered services within this API. Click on the '**Detect common objects in images**' option by clicking on the '**Try it out**' or the frame.

12. A webpage will open to preview the service.

    You may also wish to try the service by selecting an existing image or uploading an image from your local machine. This will trigger the service to be processed and then the JSON results will be shown on the right-hand side.

    Click on the '**Use the REST API**' option which will open the Computer Vision API documentation.



Recognize products on shelves
Preview
Identify products on shelves, gaps in product availability, and compliance for planograms.

Customize models with images
Preview
Create custom image classification and object detection models with images using Vision Studio and Azure ML.

Search photos with image retrieval
Preview
Retrieve specific moments within your photo album. For example, you can search for: a wedding you attended last summer, your pet, or your favorite city.

Add dense captions to images
Preview
Generate human-readable captions for all important objects detected in your image.

Remove backgrounds from images
Preview
Easily remove the background and preserve foreground elements in your image.

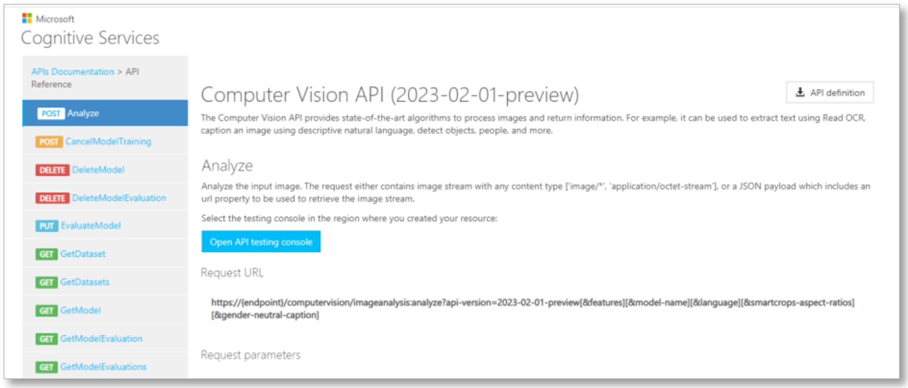
Add captions to images
Preview
Generate a human-readable sentence that describes the content of an image.

Detect common objects in images
Preview
Recognize the location of objects of interest in an image and assign them a label.

Extract common tags from images
Preview
Use an AI model to automatically assign one or more labels to an image.

Detect sensitive content in images
Detect sensitive content in images so you can moderate their usage in your applications.

Create smart-cropped images
Preview
Create cropped image thumbnails based on the key areas of a larger image.



Computer Vision API (2023-02-01-preview)

The Computer Vision API provides state-of-the-art algorithms to process images and return information. For example, it can be used to extract text using Read OCR, caption an image using descriptive natural language, detect objects, people, and more.

Analyze

Analyze the input image. The request either contains image stream with any content type ['image/*', 'application/octet-stream'], or a JSON payload which includes an url property to be used to retrieve the image stream.

Select the testing console in the region where you created your resource:

Request URL

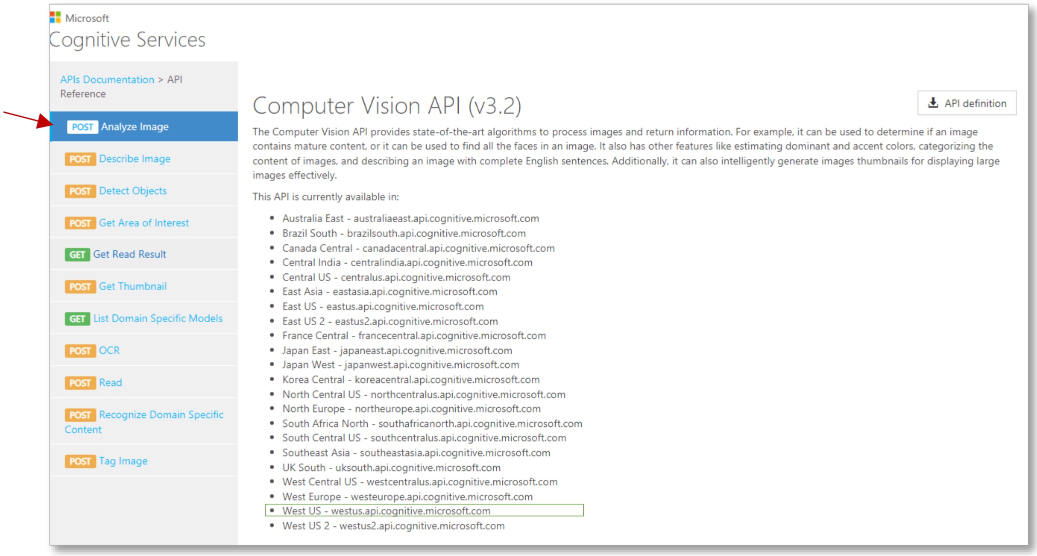
```
https://{endpoint}/computervision/imageanalysis:analyze?api-version=2023-02-01-preview[&features][&model-name][&language][&smartcrops-aspect-ratios]
[&gender-neutral-caption]
```

Request parameters

(IV) UNDERSTANDING THE SERVICE ENDPOINT URL & RUNNING CODE SAMPLES

The AI Vision API (2023-02-01-preview) is the most recent version; nonetheless, it remains in a preview phase. Therefore, we plan on using an earlier version of the API, specifically Computer Vision | Image Analysis API 3.2, which contains a range of web services like Analyze Image, Describe Image, and Detect Objects, among others.

1. Please follow this link to access the API documentation for the Computer Vision API (version 3.2).
2. The **{POST} Analyze Image** (first one) should be highlighted in a blue background by default. Choose the Analyze Image method if it is not already selected as the default selection. It should be noted that the API is accessible only in specific geographic locations, and for each region, the service endpoint varies. As an illustration, we have chosen the "**West US**" region instance, indicating that the service endpoint resides at **westus.api.cognitive.microsoft.com**. Not all web services are offered through all regions.



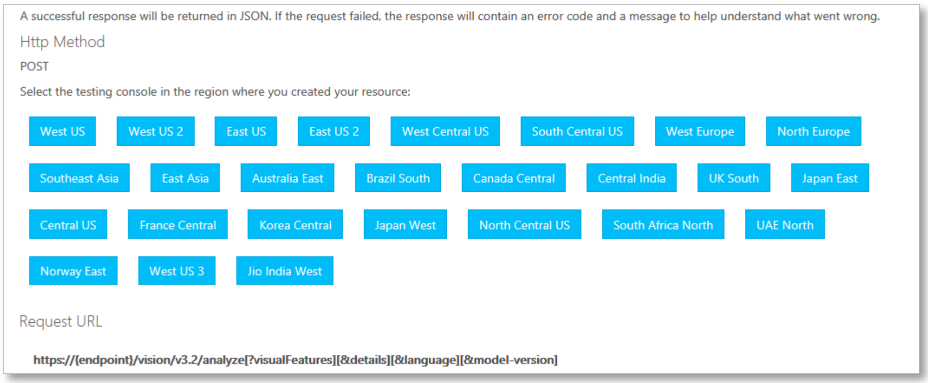
Computer Vision API (v3.2)

The Computer Vision API provides state-of-the-art algorithms to process images and return information. For example, it can be used to determine if an image contains mature content, or it can be used to find all the faces in an image. It also has other features like estimating dominant and accent colors, categorizing the content of images, and describing an image with complete English sentences. Additionally, it can also intelligently generate images thumbnails for displaying large images effectively.

This API is currently available in:

- Australia East - australiaeast.api.cognitive.microsoft.com
- Brazil South - brazilsouth.api.cognitive.microsoft.com
- Canada Central - canadacentral.api.cognitive.microsoft.com
- Central India - centralindia.api.cognitive.microsoft.com
- Central US - centralus.api.cognitive.microsoft.com
- East Asia - eastasia.api.cognitive.microsoft.com
- East US - eastus.api.cognitive.microsoft.com
- East US 2 - eastus2.api.cognitive.microsoft.com
- France Central - francecentral.api.cognitive.microsoft.com
- Japan East - japaneast.api.cognitive.microsoft.com
- Japan West - japanwest.api.cognitive.microsoft.com
- Korea Central - koreacentral.api.cognitive.microsoft.com
- North Central US - northcentralus.api.cognitive.microsoft.com
- North Europe - northeurope.api.cognitive.microsoft.com
- South Africa North - southafricanorth.api.cognitive.microsoft.com
- South Central US - southcentralus.api.cognitive.microsoft.com
- Southeast Asia - southeastasia.api.cognitive.microsoft.com
- UK South - uksouth.api.cognitive.microsoft.com
- West Central US - westcentralus.api.cognitive.microsoft.com
- West Europe - westeurope.api.cognitive.microsoft.com
- West US - westus.api.cognitive.microsoft.com
- West US 2 - westus2.api.cognitive.microsoft.com

3. Upon scrolling down, you will observe the presence of buttons corresponding to the various supporting locations. When a user clicks on a region button, a testing console will be displayed, enabling them to test the API by providing inputs and receiving corresponding outputs. Please select the "**West US**" button.



A successful response will be returned in JSON. If the request failed, the response will contain an error code and a message to help understand what went wrong.

Http Method

POST

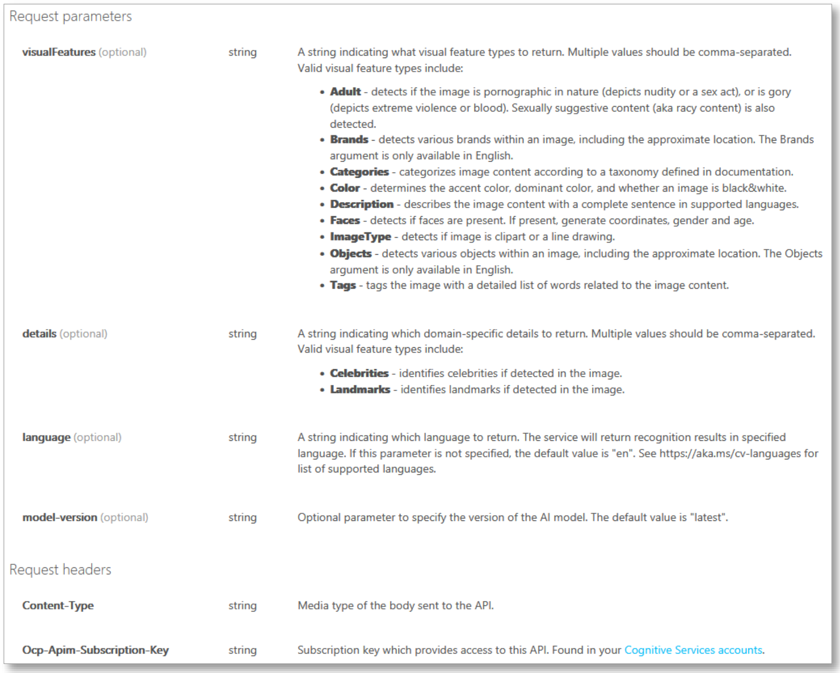
Select the testing console in the region where you created your resource:

Request URL

https://{endpoint}/vision/v3.2/analyze[?visualFeatures][&details][&language][&model-version]

It is important to acknowledge that the "analyze" service within the Computer Vision API, specifically the Analyze Image, is supports the HTTP POST method. This implies that the request URL must establish a connection using the HTTP POST method. It is important to take note of the Request URL as it provides guidance on how to properly create the URL for making a request.

4. Proceed by scrolling downwards to review the request parameters, noting that certain parameters are optional. It is important to familiarize oneself with the parameters that are provided by the method of this API. It should be noted that the request headers require the inclusion of two elements:
   a. **Content-Type**
   b. **Ocp-Apim-Subscription-Key**



Request parameters

visualFeatures (optional) string A string indicating what visual feature types to return. Multiple values should be comma-separated. Valid visual feature types include:

- **Adult** - detects if the image is pornographic in nature (depicts nudity or a sex act), or is gory (depicts extreme violence or blood). Sexually suggestive content (aka racy content) is also detected.
- **Brands** - detects various brands within an image, including the approximate location. The Brands argument is only available in English.
- **Categories** - categorizes image content according to a taxonomy defined in documentation.
- **Color** - determines the accent color, dominant color, and whether an image is black&white.
- **Description** - describes the image content with a complete sentence in supported languages.
- **Faces** - detects if faces are present. If present, generate coordinates, gender and age.
- **ImageType** - detects if image is clipart or a line drawing.
- **Objects** - detects various objects within an image, including the approximate location. The Objects argument is only available in English.
- **Tags** - tags the image with a detailed list of words related to the image content.

details (optional) string A string indicating which domain-specific details to return. Multiple values should be comma-separated. Valid visual feature types include:

- **Celebrities** - identifies celebrities if detected in the image.
- **Landmarks** - identifies landmarks if detected in the image.

language (optional) string A string indicating which language to return. The service will return recognition results in specified language. If this parameter is not specified, the default value is "en". See https://aka.ms/cv-languages for list of supported languages.

model-version (optional) string Optional parameter to specify the version of the AI model. The default value is "latest".

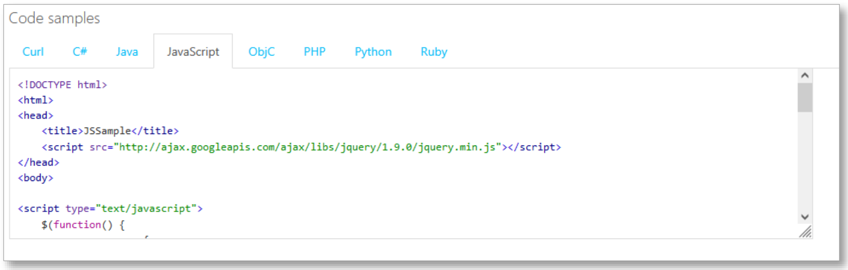
Request headers

Content-Type string Media type of the body sent to the API.

Ocp-Apim-Subscription-Key string Subscription key which provides access to this API. Found in your Cognitive Services accounts.

5. Upon scrolling below, you will be able to observe instances of the request body, alongside the option to utilize the console for the purpose of method testing. The test console also offers a sample response. You may wish to explore this testing console and acquaint yourself with the various aspects of the process, including its features, input/output arguments, output format, and possible errors.

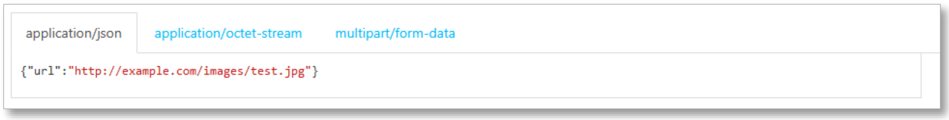
6. Upon navigating to the bottom of the page, you will encounter a collection of code samples written in several programming languages. Choose **JavaScript** as the preferred programming language.



Code samples

```
<!DOCTYPE html>
<html>
<head>
    <title>JSSample</title>
    <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.9.0/jquery.min.js"></script>
</head>
<body>

<script type="text/javascript">
    $(function() {
```

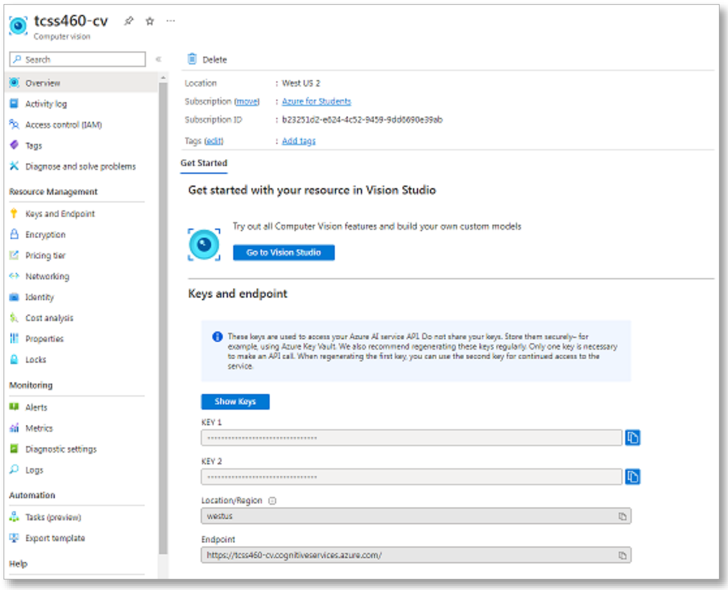
7. To begin, please replicate the above code and employ a code editor to **create a new file** on your local machine named "**tcss460-cv.html**". Next, proceed to **insert the JavaScript code** that you have previously copied from the designated section containing code examples in Step 6.
8. **In line 23**, it is important to note that the header parameter "Content-Type" is configured to "application/json". According to the description of the API, when utilizing this specific content-type, the API requires the provision of a URL corresponding to the image that is intended to be analyzed.



```
{"url":"http://example.com/images/test.jpg"}
```

In the event that the user selects the "application/octet-stream" option, the API anticipates the inclusion of the image's binary data within the request body. This entails uploading an image file to the API. Currently, the "**application/json**" content-type will be employed.

9. It is necessary to implement slight alterations to the code. Navigate to line 24 and substitute the placeholder "**{subscription key}**" with the unique key generated by the instance of the service. In order to determine the key, it is necessary to navigate to the instance information page and thereafter select the option labeled "Keys and Endpoint." By default, a pair of keys is generated. Let us replicate or copy "**KEY 1**" to the clipboard.
10. **Save** the file.
11. Then, access the file **tcss460-cv.html**, via a **web browser**. The file can be accessed by utilizing the file explorer and performing a double-click action on it. Alternatively, one can ascertain the specific location of the file on their computer and thereafter copy and paste the corresponding address into the browser's address bar.



Location : West US 2
Subscription (move) : Azure for Students
Subscription ID : b23251d2-e624-4c52-9459-9dd0690e39ab
Tags (edit) : Add tags

Get started with your resource in Vision Studio

Try out all Computer Vision features and build your own custom models

Keys and endpoint

These keys are used to access your Azure AI service API. Do not share your keys. Store them securely– for example, using Azure Key Vault. We also recommend regenerating these keys regularly. Only one key is necessary to make an API call. When regenerating the first key, you can use the second key for continued access to the service.

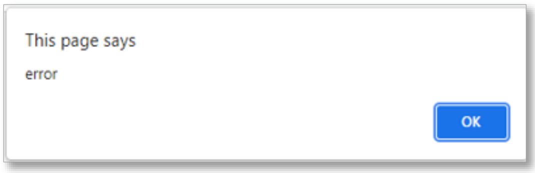
KEY 1: ••••••••••••••••••••••••••••••••

KEY 2: ••••••••••••••••••••••••••••••••

Location/Region: westus

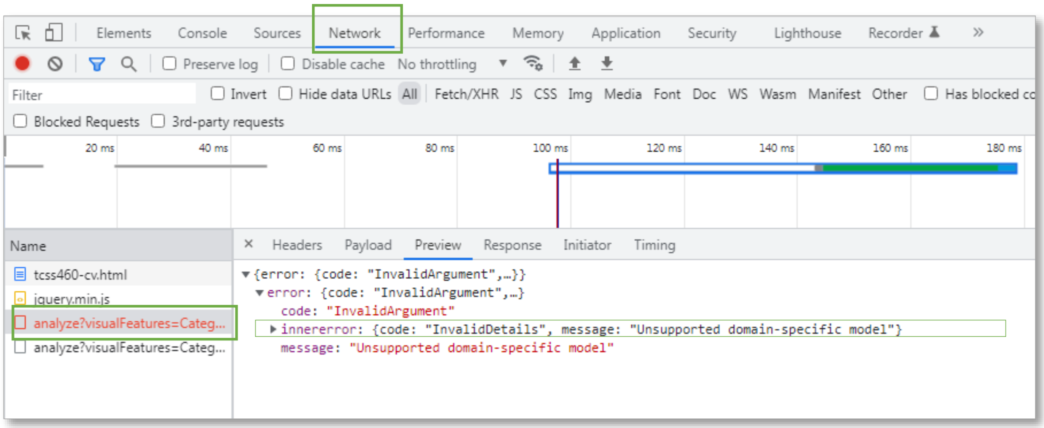
Endpoint: https://tcss460-cv.cognitiveservices.azure.com/

12. You should receive a pop-up window with "**error**" keyword/message. Press **OK**.


This page says

error

13. Then, access the developer tools on your browser (usually by pressing F12 key on your keyboard). Then, **click** on the **Network** tab. Press the **F5** key to **refresh** the page. You should see the resources consumed in this page. Press **OK** to make the pop-up go away and then click on the **line highlighted in red**. Select the **Preview** tab.



tcss460-cv.html
jquery.min.js
analyze?visualFeatures=Categ...
analyze?visualFeatures=Categ...

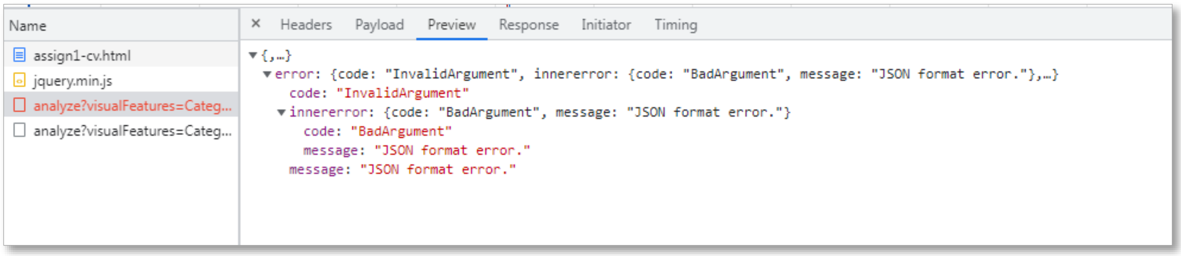
```
▼{error: {code: "InvalidArgument",…}}
  ▼error: {code: "InvalidArgument",…}
     code: "InvalidArgument"
   ▶innererror: {code: "InvalidDetails", message: "Unsupported domain-specific model"}
     message: "Unsupported domain-specific model"
```

The issue arises from the inclusion of the "**details**" parameter within the params collection of the JavaScript code. As to the API documentation, the usage of "details" denotes the exact details pertaining to a certain domain that are to be retrieved. The initial value is assigned as "{string}". The API documentation provides users with two distinct alternatives, namely "**Celebrities**" and "**Landmarks**". Therefore, it is necessary to modify the value of the params property, namely the "**details**" parameter, in order to incorporate appropriate parameters. Please modify the value for "details" on line 14 in the following manner.

```
            "details": "Celebrities,Landmarks",
```

Ensure that there are no spaces following the "," in the value as adding a space causes an error!

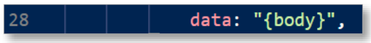
14. Press the **F5** key to **refresh** the page. You should see the resources consumed in this page. Press **OK** to make the pop-up go away and then click on the **line highlighted in red**. Select the **Preview** tab. Another error should appear for an InvalidArgument. However, this time it should be a **BadArgument** error.


Name
assign1-cv.html
jquery.min.js
analyze?visualFeatures=Categ...
analyze?visualFeatures=Categ...
× Headers Payload Preview Response Initiator Timing

```
▼{,…}
  ▼error: {code: "InvalidArgument", innererror: {code: "BadArgument", message: "JSON format error."},…}
     code: "InvalidArgument"
    ▼innererror: {code: "BadArgument", message: "JSON format error."}
       code: "BadArgument"
       message: "JSON format error."
     message: "JSON format error."
```

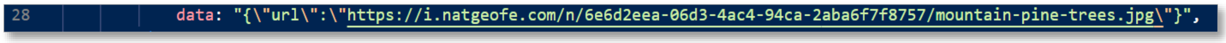
It is important to acknowledge that the error code being encountered is denoted as **InvalidArgument**, with the accompanying message indicating a JSON format error. This issue arises due to the lack of specification on the image to be analyzed. Given that we are utilizing the application/json content-type, it is imperative that we establish the precise URL of the image that we intend to analyze. Please return to the code editor and make the necessary modifications to line 28 in the following manner.

→ replace {body} with ***{"url":"http://some address to an image file"}***



```
28            data: "{body}",
```

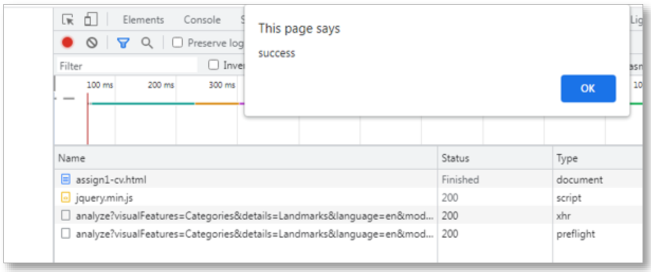
→ with **data: "{\"url\":\"https://i.natgeofe.com/n/6e6d2eea-06d3-4ac4-94ca-2aba6f7f8757/mountain-pine-trees.jpg\"}",**



```
28            data: "{\"url\":\"https://i.natgeofe.com/n/6e6d2eea-06d3-4ac4-94ca-2aba6f7f8757/mountain-pine-trees.jpg\"}",
```

Note that the "\" are used to escape the double quotes in the above string in JavaScript. From Step 8, notice that the format must be of the form: ***{"url":"http://some address to an image file"}***

15. **Save** the file. Go back to the browser and **refresh** the page for this file. You should now see a success message.

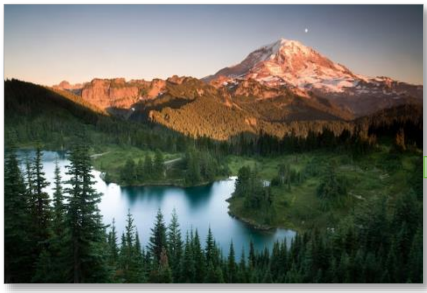


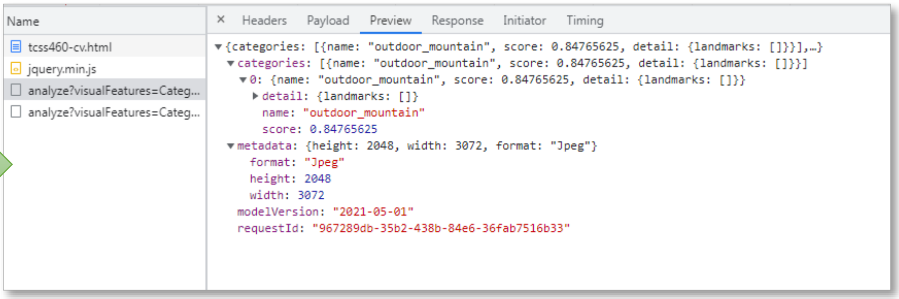
This page says

success

Name	Status	Type
assign1-cv.html	Finished	document
jquery.min.js	200	script
analyze?visualFeatures=Categories&details=Landmarks&language=en&mod...	200	xhr
analyze?visualFeatures=Categories&details=Landmarks&language=en&mod...	200	preflight

16. Click on the **analyze?visualFeatures=Categories**…and you should now see the response message content from the Computer Vision API!




Name
tcss460-cv.html
jquery.min.js
analyze?visualFeatures=Categ...
analyze?visualFeatures=Categ...
× Headers Payload Preview Response Initiator Timing

```
▼{categories: [{name: "outdoor_mountain", score: 0.84765625, detail: {landmarks: []}}],…}
  ▼categories: [{name: "outdoor_mountain", score: 0.84765625, detail: {landmarks: []}}]
    ▼0: {name: "outdoor_mountain", score: 0.84765625, detail: {landmarks: []}}
      ▶detail: {landmarks: []}
       name: "outdoor_mountain"
       score: 0.84765625
  ▼metadata: {height: 2048, width: 3072, format: "Jpeg"}
     format: "Jpeg"
     height: 2048
     width: 3072
   modelVersion: "2021-05-01"
   requestId: "967289db-35b2-438b-84e6-36fab7516b33"
```

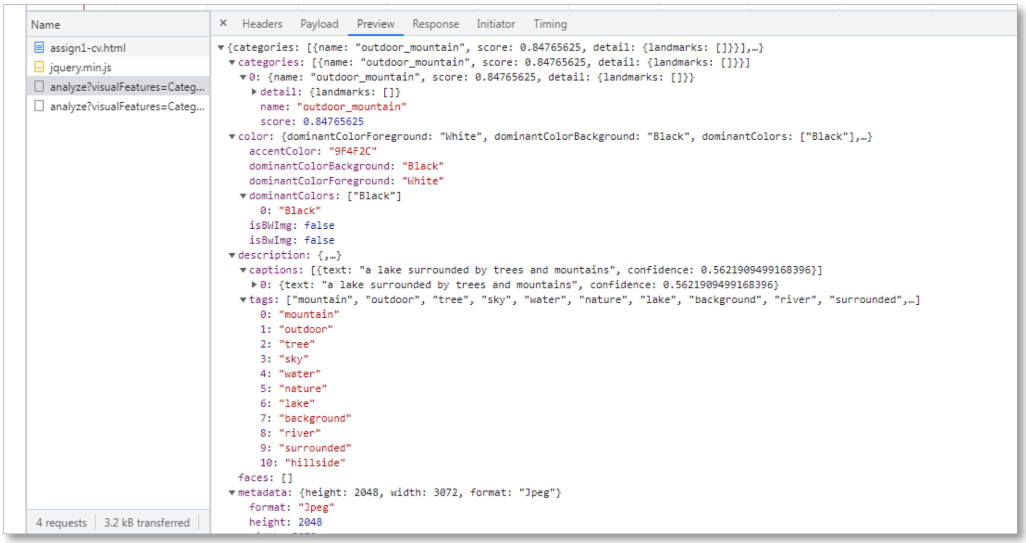
It is worth noting that the API successfully recognized the content of the image and categorized it as "**outdoor_mountain**" with a confidence score of 84.8%. Furthermore, the web service successfully determined the dimensions (height and width) as well as the format of the image. The manipulation of data from the response to HTML components on the page can be managed by utilizing jQuery or React. The JSON object output can be copied from the Response tab and further formatted using an online JSON formatter tool.

17. In step 4, the **VisualFeatures** has many options. We only used **Categories**! Go to line 13 in the code and let's add additional visual features that we wish to analyze in this image. Append the line to be as follows:

```
"visualFeatures": "Categories,Color,Description,Faces,Objects,Tags",
```

Do not leave spaces before or after the commas! The API treats the space as part of the name of the feature (e.g., fails to trim the feature name) and you will get an error!

18. **Save** the file. Go to your browser and **refresh** the page. You will see more details in the response



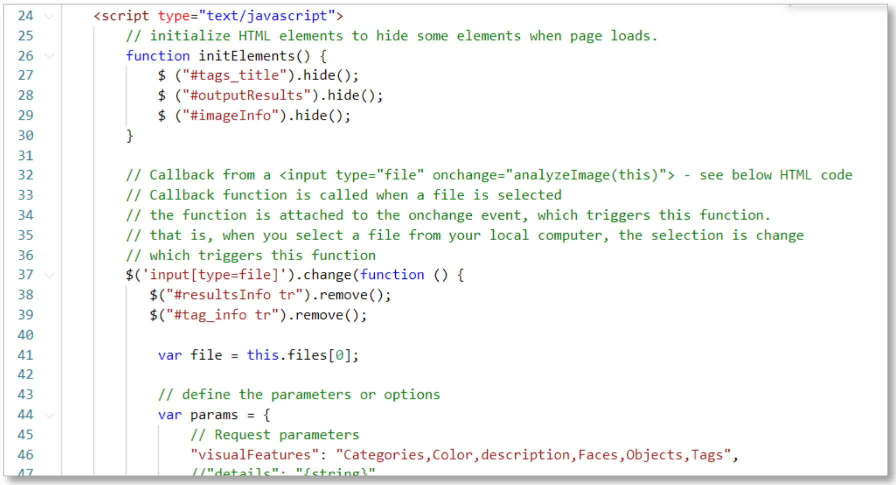
```
▼{categories: [{name: "outdoor_mountain", score: 0.84765625, detail: {landmarks: []}}],…}
  ▼categories: [{name: "outdoor_mountain", score: 0.84765625, detail: {landmarks: []}}]
    ▼0: {name: "outdoor_mountain", score: 0.84765625, detail: {landmarks: []}}
      ▶detail: {landmarks: []}
       name: "outdoor_mountain"
       score: 0.84765625
  ▼color: {dominantColorForeground: "White", dominantColorBackground: "Black", dominantColors: ["Black"],…}
     accentColor: "9F4F2C"
     dominantColorBackground: "Black"
     dominantColorForeground: "White"
    ▼dominantColors: ["Black"]
       0: "Black"
     isBWImg: false
     isBwImg: false
  ▼description: {,…}
    ▼captions: [{text: "a lake surrounded by trees and mountains", confidence: 0.5621909499168396}]
      ▶0: {text: "a lake surrounded by trees and mountains", confidence: 0.5621909499168396}
    ▼tags: ["mountain", "outdoor", "tree", "sky", "water", "nature", "lake", "background", "river", "surrounded",…]
       0: "mountain"
       1: "outdoor"
       2: "tree"
       3: "sky"
       4: "water"
       5: "nature"
       6: "lake"
       7: "background"
       8: "river"
       9: "surrounded"
       10: "hillside"
   faces: []
  ▼metadata: {height: 2048, width: 3072, format: "Jpeg"}
     format: "Jpeg"
     height: 2048
```

PART C » UPLOADING FILES & CUSTOMIZING OUTPUT USING CASCADING STYLESHEETS (CSS)

In Part B, you learned how to pass a URL of an image to the Computer Vision API. In this section, we would like to use a HTML form to upload an image file from a local machine to the Azure Computer Vision API for analysis. In addition, we use a cascading style sheet (CSS) to make the output look more professional. I will be using the following bootstrap theme: https://bootswatch.com/materia

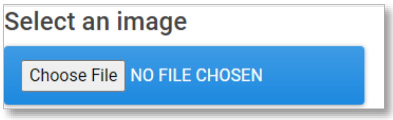
1. **Download** the **tcss460-cv-upload.html** file via Canvas to your local machine (Canvas | Module 8 | Microsoft Azure Example).

2. **Open** the HTML file through a code editor and **review the source code** and documentation to understand how it works.

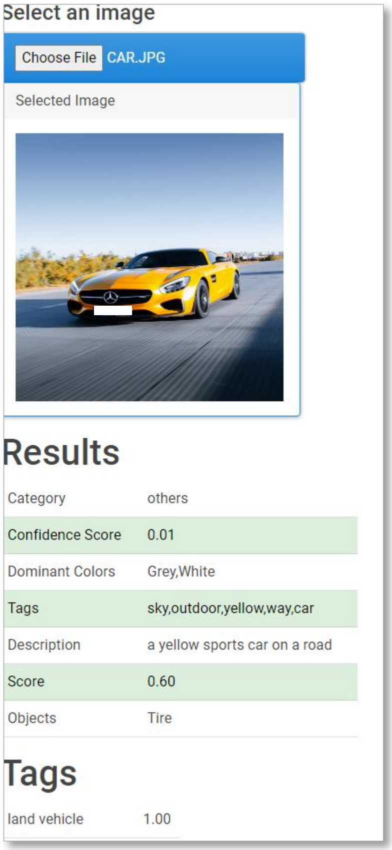


```
<script type="text/javascript">
    // initialize HTML elements to hide some elements when page loads.
    function initElements() {
        $ ("#tags_title").hide();
        $ ("#outputResults").hide();
        $ ("#imageInfo").hide();
    }

    // Callback from a <input type="file" onchange="analyzeImage(this)"> - see below HTML code
    // Callback function is called when a file is selected
    // the function is attached to the onchange event, which triggers this function.
    // that is, when you select a file from your local computer, the selection is change
    // which triggers this function
    $('input[type=file]').change(function () {
       $("#resultsInfo tr").remove();
       $("#tag_info tr").remove();

        var file = this.files[0];

        // define the parameters or options
        var params = {
            // Request parameters
            "visualFeatures": "Categories,Color,description,Faces,Objects,Tags",
            //"details": "{string}"
```

3. **Run** the file using the browser.
4. **Download** few images from the web for testing.
5. Click on the **Choose File** button and select one of the images you have downloaded


Select an image

6. Upon **opening the file**, it will be automatically submitted to the Azure Computer Vision API. Subsequently, the output outcomes derived from the answer message will be automatically assigned to corresponding HTML components.
7. **Conduct a series of experiments** utilizing various photos. The investigation aims to analyze and evaluate the source code, as well as verify the precision and correctness of the outcomes obtained from the image processing procedure utilizing the Azure Computer Vision API. Furthermore, it is imperative to analyze the output generated by the API by utilizing the F12 functionality available on your web browser.


Select an image

Results

Category	others
Confidence Score	0.01
Dominant Colors	Grey,White
Tags	sky,outdoor,yellow,way,car
Description	a yellow sports car on a road
Score	0.60
Objects	Tire

Tags

land vehicle	1.00

Congratulations!

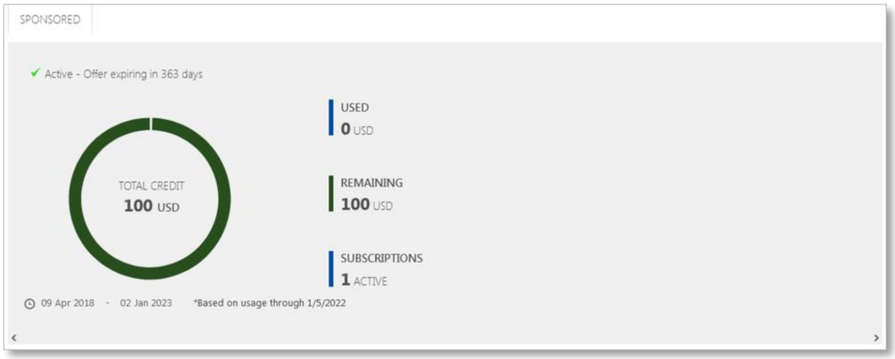
You have successfully used HTML, JavaScript and CSS to build a client frontend that connects and consumes the Azure Computer Vision API!

Important: Always Check and Monitor Your Azure Subscription Credit

It is recommended to optimize the utilization of Azure during the quarter in order to reserve an adequate amount of credit for assignments, quizzes, and the project. It is advisable to regularly examine the Azure subscription in order to ascertain the level of utilization of the $100 credit.

To check your Azure account balance:

1. Go to https://www.microsoftazuresponsorships.com/Balance
2. Check your subscription and account balance


SPONSORED

✓ Active - Offer expiring in 363 days

TOTAL CREDIT 100 USD

USED 0 USD

REMAINING 100 USD

SUBSCRIPTIONS 1 ACTIVE

09 Apr 2018 - 02 Jan 2023 *Based on usage through 1/5/2022

3. Ensure that you use the Free Tier for any of the services we use throughout the quarter, when available.