

Instalar las dependencias necesarias

Instala las librerías necesarias para interactuar con el SDK de Azure AI Vision y manejar imágenes (como matplotlib y Pillow):

```
# Instalar Azure AI Vision SDK y otras dependencias
```

```
!pip install azure-ai-vision-imageanalysis==1.0.0b3 matplotlib pillow
```

Cargar las claves de autenticación

En Colab, podemos usar archivos de configuración para guardar el endpoint y la key de Azure AI Vision de manera segura. Crea un archivo .env o define las variables directamente en tu script:

```
# Definir el endpoint y la clave directamente
```

```
import os
```

```
os.environ["AZURE_VISION_ENDPOINT"] = "tu_endpoint_aqui"
```

```
os.environ["AZURE_VISION_KEY"] = "tu_key_aqui"
```

Autenticación y análisis de imágenes

A continuación, se autentica el cliente de Azure AI Vision y se analiza una imagen, extrayendo características como descripciones y etiquetas.

```
from azure.ai.vision.imageanalysis import ImageAnalysisClient
```

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

```
from azure.core.credentials import AzureKeyCredential
```

```
# Autenticarse en el servicio Azure AI Vision
```

```
endpoint = os.getenv("AZURE_VISION_ENDPOINT")
```

```
key = os.getenv("AZURE_VISION_KEY")
```

```
client = ImageAnalysisClient(endpoint=endpoint, credential=AzureKeyCredential(key))
```

```
# Definir la imagen a analizar
```

```
image_url = 'https://raw.githubusercontent.com/MicrosoftLearning/mslearn-ai-vision/main/Labfiles/01-analyze-images/Python/image-analysis/images/street.jpg'
```

```
# Definir las características a extraer
```

```
visual_features = [  
    VisualFeatures.CAPTION,  
    VisualFeatures.TAGS,  
    VisualFeatures.OBJECTS,  
    VisualFeatures.PEOPLE  
]
```

```
# Realizar la llamada al servicio de análisis de imágenes
```

```
result = client.analyze(image_url, visual_features=visual_features)
```

```
# Mostrar resultados
```

```
if result.caption:
```

```
    print(f"Caption: '{result.caption.text}' with confidence {result.caption.confidence * 100:.2f}%")
```

```
if result.tags:
```

```
    print("\nTags detected:")
```

```
    for tag in result.tags.list:
```

```
        print(f"Tag: '{tag.name}' with confidence {tag.confidence * 100:.2f}%")
```

Visualización de los resultados

Para visualizar los resultados de objetos detectados en la imagen, puedes usar Pillow y matplotlib:

```
from PIL import Image, ImageDraw
```

```
import matplotlib.pyplot as plt
```

```
# Abrir la imagen
```

```

image = Image.open("mslearn-ai-vision/Labfiles/01-analyze-images/Python/image-
analysis/images/street.jpg")

draw = ImageDraw.Draw(image)

# Dibujar los cuadros delimitadores de los objetos detectados

if result.objects:

    print("\nObjects detected:")

    for detected_object in result.objects.list:

        print(f"{detected_object.tags[0].name} with confidence {detected_object.confidence *
100:.2f}%")

        r = detected_object.bounding_box

        draw.rectangle(((r.x, r.y), (r.x + r.width, r.y + r.height)), outline="cyan", width=3)

# Mostrar la imagen con los objetos detectados

plt.imshow(image)

plt.axis('off')

plt.show()

```

Eliminar el fondo de una imagen

Para eliminar el fondo o generar un **matte foreground**, puedes realizar una llamada a la API de análisis con un modo específico:

```

import requests

# Definir la API para eliminar el fondo

api_version = "2023-02-01-preview"

mode = "backgroundRemoval" # Opción: "foregroundMatting"

# URL de la imagen a procesar

image_url = 'https://raw.githubusercontent.com/MicrosoftLearning/mslearn-ai-
vision/main/Labfiles/01-analyze-images/Python/image-analysis/images/street.jpg'

# Llamada a la API para remover el fondo

```

```
url = f"{endpoint}computervision/imageanalysis:segment?api-  
version={api_version}&mode={mode}"
```

```
headers = {  
    "Ocp-Apim-Subscription-Key": key,  
    "Content-Type": "application/json"  
}
```

```
body = {  
    "url": image_url,  
}
```

```
# Hacer la petición a la API
```

```
response = requests.post(url, headers=headers, json=body)
```

```
# Guardar la imagen resultante
```

```
with open("background_removed.png", "wb") as f:
```

```
    f.write(response.content)
```

```
# Mostrar la imagen con fondo removido
```

```
background_removed = Image.open("background_removed.png")
```

```
plt.imshow(background_removed)
```

```
plt.axis('off')
```

```
plt.show()
```