**Endpunkt Anleitung**

**Endpunkt-URL & Authentifizierung**

Der AML-Endpunkt erwartet eine HTTP-POST-Anfrage mit einem Base64-kodierten Bild im JSON-Format. Die Authentifizierung erfolgt über einen API-Schlüssel (Bearer-Token).

**Format des API-Schlüssels:**  
kbnW43lXpC3S329NCgp0gazA9VxSxFXp

**Erwartetes Format der Endpunkt-URL:**

https://<Endpoint-Name>.westeurope.inference.ml.azure.com/score

Hinweis: <Endpoint-Name> durch tatsächlichen Endpunktnamen ersetzen

**HTTP-Header:**Definition der Metadaten der Anfrage

headers = {

"Content-Type": "application/json",

"Authorization": "Bearer <API-Schlüssel>"

}

**Request-Body:**Übergabe des Base64-kodierten Bildes

{

"image": "<BASE64-kodiertes Bild>"

}

**Fehlerbehandlung & Rückgabecodes**

Falls Probleme auftreten, gibt der Endpunkt folgende **HTTP-Statuscodes** zurück:

|  |  |
| --- | --- |
| **Statuscode** | **Bedeutung** |
| **200 OK** | Erfolgreiche Analyse, Ergebnis im JSON-Format |
| **400 Bad Request** | Ungültige oder fehlende Eingabe |
| **408 Request Timeout** | Kein Bild in der Anfrage enthalten |
| **415 Unsupported Media Type** | Bildformat wird nicht unterstützt (nur JPG, PNG, BMP, WEBP, TIFF) |
| **424 Failed Dependency** | YOLO-Modell nicht geladen |
| **500 Internal Server Error** | Interner Fehler (Modell- oder OCR-Verarbeitung fehlgeschlagen) |

**Beispiel Endpunktanfrage - Minimal Python Code:**

import json

import base64

import urllib.request

# Azure Machine Learning (AML) Endpoint-URL für Modellinferenz

url = "https://end.westeurope.inference.ml.azure.com/score"

# API-Schlüssel für die Authentifizierung beim AML-Endpunkt

api\_key = 'kbnW43lXpC3S329NCgp0gazA9VxSxESp'

def encode\_image(image\_path):

"""

Liest eine Bilddatei ein und kodiert sie in Base64-Format.

Dies ist erforderlich, da viele ML-Endpunkte Bilder in Base64-codierter Form erwarten.

:param image\_path: Pfad zur Bilddatei

:return: Base64-codierter String der Bilddatei

"""

with open(image\_path, "rb") as image\_file:

return base64.b64encode(image\_file.read()).decode("utf-8")

def send\_request(encoded\_image):

"""

Sendet eine HTTP-POST-Anfrage an den AML-Endpunkt mit einem Base64-codierten Bild.

Die Antwort des Modells wird als JSON-Objekt zurückgegeben.

:param encoded\_image: Base64-codierter String der Bilddatei

:return: JSON-Antwort des Modells

"""

data = json.dumps({"image": encoded\_image}).encode("utf-8") # Anfrage-Daten im JSON-Format kodieren

headers = {

"Content-Type": "application/json", # Definiert den Inhaltstyp als JSON

"Authorization": f"Bearer {api\_key}" # Autorisierung mittels API-Schlüssel

}

req = urllib.request.Request(url, data=data, headers=headers, method="POST") # HTTP-Request vorbereiten

with urllib.request.urlopen(req) as response: # Anfrage senden und Antwort empfangen

return json.loads(response.read().decode("utf-8")) # Antwort in JSON umwandeln

# Bild wird in Base64-Format konvertiert, um es an den Endpunkt zu senden

encoded\_image = encode\_image("picture\_test.png")

# Anfrage an den Endpunkt senden und die Vorhersage abrufen

response\_json = send\_request(encoded\_image)

# Ausgabe der Modellantwort

print(response\_json)