**Documentacion de Trabajo e clase**

import cv2

import requests

import time

from clarifai\_grpc.channel.clarifai\_channel import ClarifaiChannel

from clarifai\_grpc.grpc.api import resources\_pb2, service\_pb2, service\_pb2\_grpc

from clarifai\_grpc.grpc.api.status import status\_code\_pb2

# Configuración de Clarifai (necesitarás una API key)

CLARIFAI\_API\_KEY = 'Inserte API key'  # Reemplaza con tu API key de Clarifai

USER\_ID = '2jl42efzat1y'

APP\_ID = 'Reconocimiento-de-personas'

MODEL\_ID = 'general-image-recognition'

MODEL\_VERSION\_ID = 'aa7f35c01e0642fda5cf400f543e7c40'

def tomar\_foto():

    # Inicializar la cámara

    cap = cv2.VideoCapture(0)

    if not cap.isOpened():

        print("No se pudo abrir la cámara")

        return None

    print("Presiona 's' para tomar la foto o 'q' para salir")

    while True:

        ret, frame = cap.read()

        if not ret:

            print("Error al capturar el frame")

            break

        cv2.imshow('Cámara - Presiona "s" para tomar foto', frame)

        key = cv2.waitKey(1)

        if key == ord('s'):

            # Guardar la imagen temporalmente

            filename = 'foto\_capturada.jpg'

            cv2.imwrite(filename, frame)

            print(f"Foto guardada como {filename}")

            break

        elif key == ord('q'):

            print("Saliendo sin tomar foto")

            filename = None

            break

    # Liberar recursos

    cap.release()

    cv2.destroyAllWindows()

    return filename

def analizar\_imagen\_con\_clarifai(image\_path):

    channel = ClarifaiChannel.get\_grpc\_channel()

    stub = service\_pb2\_grpc.V2Stub(channel)

    metadata = (('authorization', f'Key {CLARIFAI\_API\_KEY}'),)

    with open(image\_path, 'rb') as f:

        file\_bytes = f.read()

    request = service\_pb2.PostModelOutputsRequest(

        user\_app\_id=resources\_pb2.UserAppIDSet(user\_id=USER\_ID, app\_id=APP\_ID),

        model\_id=MODEL\_ID,

        version\_id=MODEL\_VERSION\_ID,

        inputs=[

            resources\_pb2.Input(

                data=resources\_pb2.Data(

                    image=resources\_pb2.Image(

                        base64=file\_bytes

                    )

                )

            )

        ])

    response = stub.PostModelOutputs(request, metadata=metadata)

    if response.status.code != status\_code\_pb2.SUCCESS:

        print(f"Error en la solicitud: {response.status.description}")

        return None

    return response.outputs[0].data.concepts

def interpretar\_resultados(concepts):

    es\_persona = False

    otros\_objetos = []

    conceptos\_persona = ['person', 'people', 'adult', 'man', 'woman', 'child', 'portrait']

    for concept in concepts:

        if concept.name in conceptos\_persona and concept.value > 0.85:

            es\_persona = True

        elif concept.value > 0.7:

            otros\_objetos.append((concept.name, concept.value))

    otros\_objetos.sort(key=lambda x: x[1], reverse=True)

    return es\_persona, otros\_objetos[:5]

def main():

    # Paso 1: Tomar la foto

    imagen = tomar\_foto()

    if not imagen:

        return

    # Paso 2: Analizar la imagen con Clarifai

    print("Analizando imagen...")

    conceptos = analizar\_imagen\_con\_clarifai(imagen)

    if not conceptos:

        print("No se pudo analizar la imagen")

        return

    # Paso 3: Interpretar los resultados

    es\_persona, otros\_objetos = interpretar\_resultados(conceptos)

    # Paso 4: Mostrar resultados

    if es\_persona:

        print("\n¡Se ha detectado una persona en la imagen!")

    else:

        print("\nNo se detectó una persona en la imagen. Los objetos identificados son:")

        for obj, confianza in otros\_objetos:

            print(f"- {obj} (confianza: {confianza:.2%})")

if \_\_name\_\_ == "\_\_main\_\_":

    main()

Utilizamos Clarifai sobre Google Cloud Vision por Facilidad de uso ya que ofrece modelos preentrenados listos para usar, como "general-image-recognition", con una API bastante directa. Aparte ofrece una versión gratuita. No necesita de configurar servicios adicionales ya que Clarifai, basta con una API key.

**Como funciona**

Este programa toma una foto con la cámara, la analiza usando la API de Clarifai y determina si hay una persona en la imagen. Aquí tienes un resumen breve del funcionamiento:

Captura de imagen: Usa OpenCV para acceder a la cámara. El usuario debe presionar 's' para tomar la foto.

Análisis con Clarifai: Envía la imagen capturada al modelo de reconocimiento general de Clarifai mediante gRPC y una API Key.

Interpretación de resultados: Examina las etiquetas (conceptos) devueltas por Clarifai y verifica si se detectó una persona con alta confianza (>85%).

Resultado: Informa si se detectó una persona o, en su defecto, muestra los principales objetos detectados con más del 70% de confianza.

**PROMT UTILIZADO:** Desarolla un codigo para phyton en el cual la camara de la computadora tome una foto y que mediante una api pueda identificar si lo que contiene es una persona y si no indentifique que es lo que aparece.

