UNIVERSIDAD PRIVADA DOMINGO SAVIO

FACULTAD DE INGENIERÍA

PROYECTO FORMATIVO

MATERIA: INTELIGENCIA ARTIFICIAL

DOCENTE: Ing. Jaime Zambrana Chacon

INTEGRANTES:

- **Jesus Gabriel Justiniano Enriquez**
- **Wilmer Michel Condori**
- **Alejandro López Molina**
- **Esnaider juan Aguilera**
- **Sergio Garafulic Baldiviezo**
- **Edward alexander viruez roca**

Extractor y Resolvedor de Ecuaciones

Introducción

Este proyecto tiene como objetivo desarrollar una aplicación que permite a los usuarios cargar imágenes que contengan ecuaciones matemáticas, extraer el texto de estas imágenes mediante técnicas de reconocimiento óptico de caracteres (OCR), y resolver las ecuaciones utilizando la API de OpenAI. La aplicación está diseñada con una interfaz gráfica utilizando Tkinter, facilitando su uso y acceso.

Objetivos

1. **Cargar una imagen de ecuación**: Permitir al usuario seleccionar una imagen desde su sistema.
2. **Extraer texto de la imagen**: Usar Tesseract para convertir la imagen en texto legible.
3. **Resolver la ecuación**: Utilizar OpenAI para obtener la solución de la ecuación extraída y mostrarla al usuario.

Descripción del Código

Importaciones

Se importan las librerías necesarias para el funcionamiento del proyecto:

```
import tkinter as tk
from tkinter import messagebox, filedialog
from PIL import Image, ImageTk
import pytesseract
import openai
```

Configuración de Tesseract y OpenAI

La ruta de Tesseract se establece para permitir el acceso al ejecutable, y se proporciona la clave de API de OpenAI para utilizar sus servicios:

```
pytesseract.pytesseract.tesseract_cmd = r'C:\Program Files\Tesseract-OCR\tesseract.exe'
openai.api_key = "tu_clave_de_api_aqui"  # Reemplaza con tu clave de API
```

Función cargar_imagen()

Esta función permite al usuario seleccionar una imagen y realiza las siguientes tareas:

- Carga y muestra la imagen seleccionada en el canvas.
- Extrae el texto de la imagen utilizando Tesseract.
- Muestra la ecuación extraída al usuario.
- Llama a la función resolver_ecuacion para obtener la solución de la ecuación.

```
def cargar_imagen():
    # Código para cargar la imagen, extraer el texto y resolver la ecuación
```

Función resolver_ecuacion(ecuacion)

Esta función utiliza la API de OpenAI para resolver la ecuación extraída y presenta la solución al usuario. Se maneja cualquier excepción que pueda surgir durante el proceso:

```
def cargar_imagen():
    # Código para cargar la imagen, extraer el texto y resolver la ecuación
```

Configuración de la Interfaz Gráfica

Se establece la ventana principal de la aplicación y se configuran los elementos de la interfaz, incluyendo un botón para cargar la imagen y un canvas para mostrarla:

```
root = tk.Tk()
root.title("Extractor y Resolvedor de Ecuaciones")
root.geometry("500x500")

cargar_button = tk.Button(root, text="Cargar Imagen de Ecuación", command=cargar_imagen)
cargar_button.pack(pady=10)

canvas = tk.Canvas(root, width=400, height=400)
canvas.pack(pady=20)

root.mainloop()
```

Participación y Colaboración

Durante el desarrollo del proyecto, se realizó una discusión activa en equipo, donde todos los miembros participaron en el desarrollo y la implementación del código. Se fomentó un ambiente de colaboración, permitiendo que cada persona contribuyera con ideas y sugerencias.

La utilización de la API de OpenAI fue fundamental, proporcionando una forma rápida y eficiente de resolver ecuaciones complejas. Cada miembro del equipo tuvo la oportunidad de experimentar con la API, fortaleciendo su comprensión de las capacidades de inteligencia artificial.

Resultados

La aplicación logró cargar imágenes de ecuaciones, extraer el texto con alta precisión y resolver las ecuaciones presentadas utilizando la API de OpenAI. Las soluciones se mostraron de manera efectiva al usuario, cumpliendo con los objetivos planteados.

Conclusiones

El proyecto ha demostrado ser exitoso en la creación de una herramienta útil para la extracción y resolución de ecuaciones matemáticas a partir de imágenes. La implementación de Tesseract y OpenAI ha permitido integrar eficazmente el OCR y la resolución matemática en una interfaz gráfica amigable.

Se recomienda continuar con el desarrollo de la aplicación para incluir características adicionales, como la capacidad de resolver diferentes tipos de ecuaciones y mejorar el proceso de limpieza del texto extraído. También sería beneficioso asegurar la privacidad y seguridad al manejar la clave de API de OpenAI.

Código Fuente

A continuación, se incluye el código fuente utilizado durante el proyecto:

```
import tkinter as tk

from tkinter import messagebox, filedialog

from PIL import Image, ImageTk
```

```
import pytesseract

import openai


# Configura la ruta de Tesseract

pytesseract.pytesseract.tesseract_cmd = r'C:\Program Files\Tesseract-OCR\tesseract.exe'  # Cambia la ruta si es necesario


# Proporciona tu clave de API de OpenAI

openai.api_key = "tu_clave_de_api_aqui"  # Reemplaza con tu clave de API


# Función para cargar la imagen y extraer la ecuación

def cargar_imagen():

    # Abre un cuadro de diálogo para seleccionar una imagen

    file_path = filedialog.askopenfilename(title="Selecciona una imagen", filetypes=[("Archivos de imagen", ".png;.jpg;.jpeg;.bmp")])


    if file_path:

        try:

            # Carga la imagen

            img = Image.open(file_path)

            img.thumbnail((400, 400))  # Ajusta el tamaño para mostrarla

            img_tk = ImageTk.PhotoImage(img)


            # Muestra la imagen en el canvas

            canvas.create_image(200, 200, image=img_tk)
```

```
            canvas.image = img_tk  # Mantiene una referencia a la imagen


            # Extrae el texto de la imagen (la ecuación)
            texto = pytesseract.image_to_string(img)
            messagebox.showinfo("Ecuación extraída", texto)  # Muestra la ecuación extraída


            # Resuelve la ecuación usando OpenAI
            resolver_ecuacion(texto.strip())  # Elimina espacios en blanco


        except Exception as e:
            messagebox.showerror("Error", f"Error al procesar la imagen: {str(e)}")


# Función para resolver la ecuación con OpenAI
def resolver_ecuacion(ecuacion):
    try:
        response = openai.ChatCompletion.create(
            model="gpt-3.5-turbo",  # O el modelo que prefieras
            messages=[
                {"role": "user", "content": f"Resuelve la siguiente ecuación: {ecuacion}"}
            ]
        )
        respuesta = response.choices[0].message['content']
        messagebox.showinfo("Solución", respuesta)  # Muestra la solución
```

```
    except Exception as e:
        messagebox.showerror("Error", f"Error al resolver la ecuación: {str(e)}")

# Configuración de la interfaz gráfica
root = tk.Tk()
root.title("Extractor y Resolvedor de Ecuaciones")
root.geometry("500x500")

# Botón para cargar la imagen
cargar_button = tk.Button(root, text="Cargar Imagen de Ecuación", command=cargar_imagen)
cargar_button.pack(pady=10)

# Canvas para mostrar la imagen cargada
canvas = tk.Canvas(root, width=400, height=400)
canvas.pack(pady=20)

# Iniciar la interfaz gráfica
root.mainloop()
```