



UNIVERSIDAD DEL BÍO-BÍO

16 Procesamiento y análisis de imágenes cargadas

Tutorial Procesamiento y análisis de imágenes cargadas

Estudiantes Practicantes:

Luis Pereira

Profesor:

Luis Vera

Laboratorio CIMUBB

2023-2

Procesamiento y análisis de imágenes cargadas

Importación de Bibliotecas

El código comienza importando las bibliotecas necesarias:

- 1) tkinter se importa como tk para crear la interfaz gráfica.
- 2) Se importan los módulos filedialog y messagebox para trabajar con cuadros de diálogo de archivos y mensajes emergentes.
- 3) cv2 es el módulo de OpenCV para el procesamiento de imágenes.
- 4) numpy se utiliza para operaciones numéricas y manipulación de matrices.

```
import tkinter as tk
from tkinter import filedialog
from tkinter import messagebox
import cv2
import numpy as np
```

Funciones utilizadas

- 1) **cargar_imagen():** Esta función se encarga de abrir un cuadro de diálogo para seleccionar una imagen y cargarla. La imagen se carga en escala de grises utilizando OpenCV.

```
def cargar_imagen():
    global imagen
    ruta_archivo = filedialog.askopenfilename(filetypes=[("Imágenes", "*.jpg *.jpeg *.png *.bmp")])
    if ruta_archivo:
        imagen = cv2.imread(ruta_archivo, cv2.IMREAD_GRAYSCALE) # Cargar la imagen en escala de grises
        if imagen is not None:
            messagebox.showinfo("Éxito", "Imagen cargada correctamente")
        else:
            messagebox.showerror("Error", "No se pudo cargar la imagen")
```

- 2) **analizar_imagen():** Realiza el análisis de la imagen cargada, calculando la recurrencia de intensidad y el número de celdas con valor 1 (blanco) y 0 (negro) en función de un umbral definido por el usuario

```
def analizar_imagen():
    global imagen
    if imagen is not None:
        # Realizar análisis de la matriz de intensidad
        umbral = int(numeroUmbral.get()) # Umbral de ejemplo (ajusta según tus necesidades)
        matriz_binaria = np.where(imagen > umbral, 1, 0) # Umbralización
        recurrencia = np.count_nonzero(matriz_binaria) / (matriz_binaria.shape[0] * matriz_binaria.shape[1])
        mensaje = f"Recurrencia de intensidad (umbral {umbral}): {recurrencia * 100:.2f}%\n"
        mensaje += f"Celdas con valor 1 (Blanco): {np.count_nonzero(matriz_binaria)}\n"

        mensaje += f"Celdas con valor 0 (Negro): {matriz_binaria.size - np.count_nonzero(matriz_binaria)}\n"

        # Mostrar el resultado en un cuadro de texto
        resultado_text.config(state=tk.NORMAL)
        resultado_text.delete(1.0, tk.END) # Borrar contenido anterior
        resultado_text.insert(tk.END, mensaje)
        resultado_text.config(state=tk.DISABLED)
    else:
        messagebox.showerror("Error", "Primero carga una imagen")
```

Ventana Principal

Se crea una ventana principal con el título "Analizador de Imágenes" y se establece su tamaño como 500x300 píxeles. La ventana no es redimensionable (resizable(0,0)).

```
ventana = tk.Tk()
ventana.title("Analizador de Imágenes")
ventana.geometry("500x300")
ventana.resizable(0,0)
```

Variables Globales

La variable global imagen se utiliza para almacenar la imagen cargada y se inicializa como None.

```
# Variables globales
imagen = None
```

Botones

Se crean dos botones, "Cargar Imagen" y "Analizar Imagen", que ejecutan las funciones cargar_imagen y analizar_imagen respectivamente cuando se hacen clic en ellos.

```
# Botones
btn_cargar_imagen = tk.Button(ventana, text="Cargar Imagen", command=cargar_imagen)
btn_analizar = tk.Button(ventana, text="Analizar Imagen", command=analizar_imagen)
```

Cuadro de Texto

Un cuadro de texto se crea para mostrar los resultados del análisis. Está configurado para que no se pueda editar manualmente (estado DISABLED).

```
# Cuadro de texto para mostrar el resultado
resultado_text = tk.Text(ventana, wrap=tk.WORD, height=10, width=50, state=tk.DISABLED)
```

SpinBox y Etiqueta

Un SpinBox permite al usuario seleccionar un umbral (valor entre 0 y 255) para la umbralización de la imagen. Se muestra junto con una etiqueta que indica "Umbral".

```
#SpinBox
numeroUmbral = tk.Spinbox(ventana, from_=0, to=255)
numeroUmbral.place(x=150, y=20, width=42, height=22)
LabelUmbral = tk.Label(ventana, text="Umbral: ")
LabelUmbral.place(x=100, y=20)
# Iniciar la aplicación
ventana.mainloop()
```

Instrucciones

Se agrega una etiqueta que proporciona una breve explicación de lo que hace el programa.

```
explicacion = tk.Label(ventana, text="Este programa carga la imagen como una matriz y analiza celda a celda")
explicacion.place(x=60, y=230)
```

Iniciar la Aplicación

La aplicación se inicia con ventana.mainloop(), lo que permite al usuario interactuar con la interfaz.

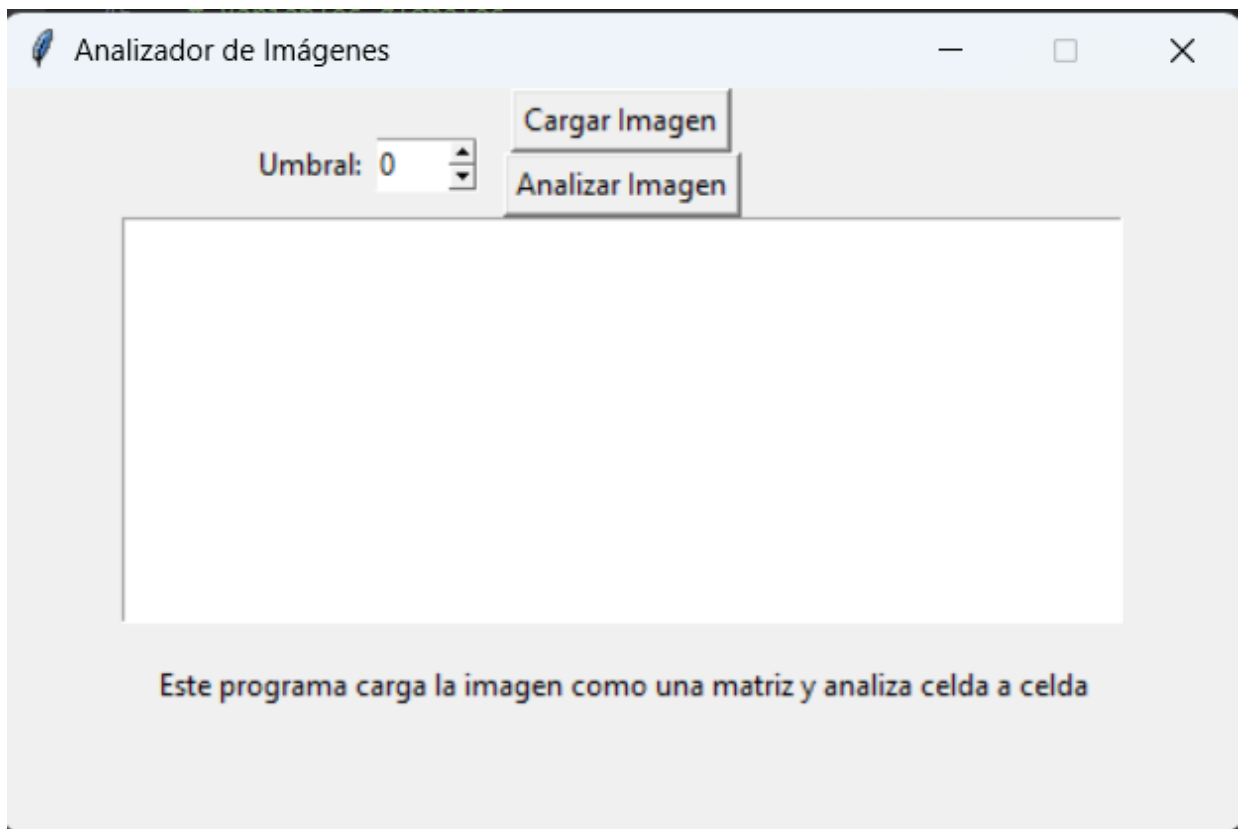
```
# Iniciar la aplicación
ventana.mainloop()
```

Funcionamiento

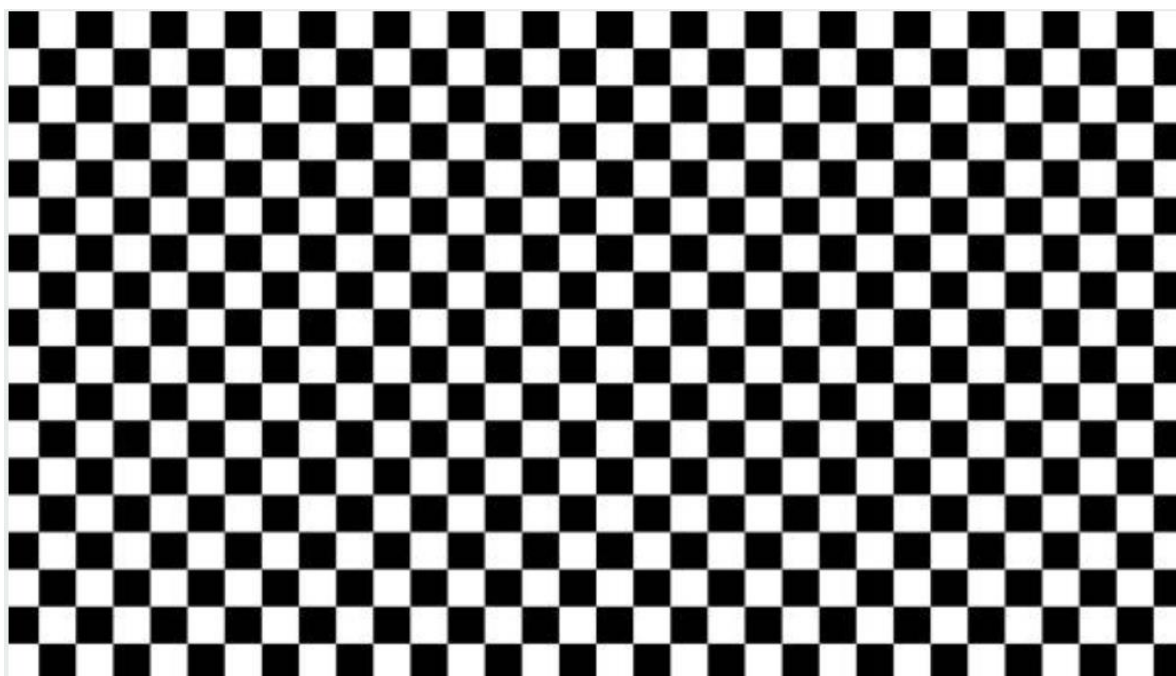
El programa permite a los usuarios cargar una imagen en escala de grises, definir un umbral y analizar la imagen para calcular la recurrencia de intensidad y mostrar el número de celdas en blanco y negro. Los resultados se presentan en un cuadro de texto en la ventana.

En resumen, este programa proporciona una interfaz simple para cargar y analizar imágenes en escala de grises, lo que puede ser útil para realizar análisis de imágenes basados en umbralización y recurrencia de intensidad.

Resultado final



En funcionamiento al cargar la siguiente imagen:



Obtenemos como resultado...

