UNIVERSIDAD DEL BÍO-BÍO

09 Tomar foto usando webcam

Tutorial Procesamiento de Imagen con webcam

Estudiantes Practicantes:
Luis Pereira

Profesor:
Luis Vera

Laboratorio CIMUBB

2023-2

Tomar foto usando webcam

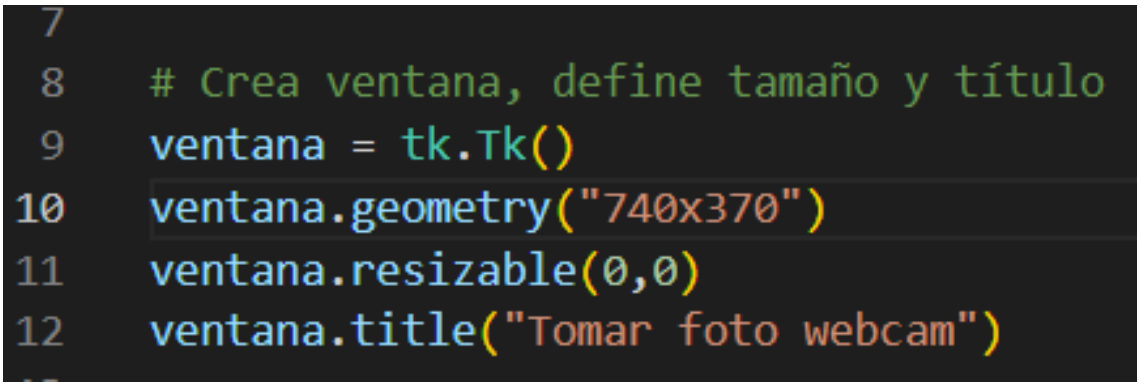
Importación de bibliotecas:

La aplicación desarrollada en Python combina las capacidades de diversas bibliotecas, como tkinter, OpenCV (cv2), Pillow (PIL), imutils y os, para proporcionar una interfaz amigable para la captura de imágenes desde una cámara web y su posterior almacenamiento en el sistema de archivos del usuario.

Creación de la ventana:

Se crea una ventana principal utilizando **tk.Tk().**

La ventana se establece con un tamaño fijo de 740x460 píxeles y se deshabilita la posibilidad de redimensionarla con **ventana.geometry("740x460"**) y **ventana.resizable(0, 0).**



Iniciar camara

Al igual que en el tutorial de iniciar la cámara web, creados dos funciones, una que se acciona con el botón “Iniciar cámara” y otra que llama la función **camara():**

```
#Funciones cámara web
def camara():
    global capture
    capture = cv2.VideoCapture(0)
    iniciar()

def iniciar():
    global capture
    if capture is not None:
        ret, frame = capture.read()
        if ret == True:
            frame = imutils.resize(frame, width=311)
            frame = imutils.resize(frame, height=241)
            ImagenCamara = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
            im = Image.fromarray(ImagenCamara)
            img = ImageTk.PhotoImage(image= im)
            LImagen.configure(image= img)
            LImagen.image = img
            LImagen.after(1,iniciar)
        else:
            LImagen.image = ""
            capture.release()
```

Capturar frame

Creamos otra función llamada Capturar que será la responsable de que se guarde un frame de la cámara y se muestre en la ventana del label, tener en cuenta que cv2 por defecto captura el frame como BGR (Blue-Green-Red) y con el método cvtColor podemos pasar el frame de escala de BGR a RGB (Red-Green-Blue) y guardamos la captura en la variable Captura.

```
#Función para tomar una foto
def Capturar():
    global Captura
    camara = capture
    return_value, image = camara.read()
    frame = imutils.resize(image, width=301)
    frame = imutils.resize(frame, height=221)
    Captura = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    im = Image.fromarray(Captura)
    img = ImageTk.PhotoImage(image= im)
    LImagenROI.configure(image= img)
    LImagenROI.image = img
```

Captura temporizada

También creamos otra función llamada capturar con temporizador que nos permite escoger un tiempo de 1 a 60 segundos en un SpinBox y tomara imágenes automáticamente de acuerdo con el tiempo que hemos escogido, también tenemos la opción de seleccionar una carpeta y darle un nombre al archivo, notar que el método after nos permite hacer una llamada recursiva a la función según el tiempo que fue seleccionado en el SpinBox.

```
def capturar_con_temporizacion():
    global ejecutar_solo_una_vez
    if ejecutar_solo_una_vez == True:
        messagebox.showwarning("Advertencia", "Se estan guardando fotogramas en un ciclo infinito, no te olvides de cerrar el programa o tu disco duro colapsara")
    ejecutar_solo_una_vez = False

    global Captura
    camara = capture
    ret, image = camara.read()
    frame = imutils.resize(image, width=301)
    frame = imutils.resize(frame, height=221)
    Captura = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    im = Image.fromarray(Captura)
    img = ImageTk.PhotoImage(image=im)
    LImagenROI.configure(image=img)
    LImagenROI.image = img
    # Programa la próxima captura
    guardar_imagen()
    ventana.after(1000 * int(numeroTemp.get()), capturar_con_temporizacion)
```

Botones

Creamos los botones para iniciar la cámara, tomar foto con temporizador, guardar imagen, seleccionar carpeta y para tomar la foto única, que con el **command** llamarán a la función que se requiera.

También se añaden 2 recuadros grises que representarán donde se ubicará la webcam y la foto que tomemos.

Se añade un label instructivo que le indica al usuario que antes de clicar en tomar foto temporizada debe seleccionar una carpeta para que se guarden las imágenes y poner nombre a la imagen que por defecto será guardado en .png.

Guardar imagen

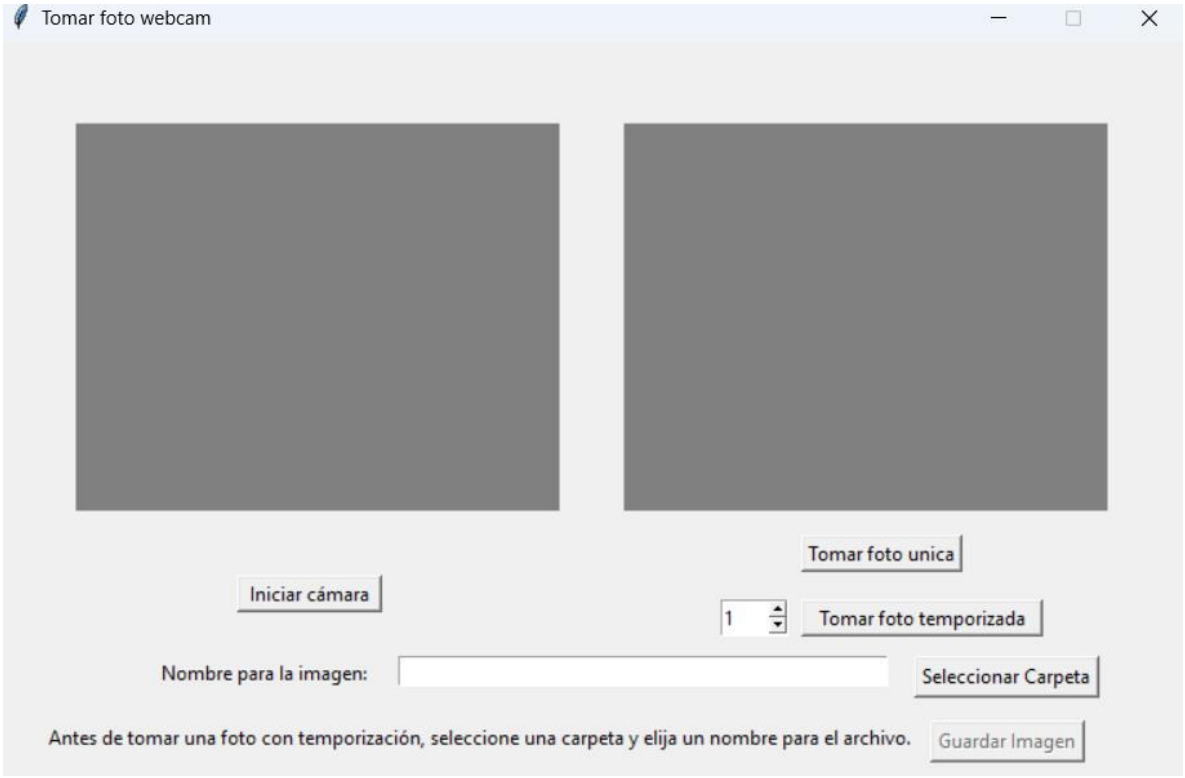
También en la función **guardar_imagen**() que habiendo previamente seleccionado una carpeta, guardara la imagen en la carpeta escogida, como se van a ir tomando fotos temporizadas y el nombre que le pusimos a la imagen no cambiara, la imagen se guardara en la carpeta seleccionada y se reemplazara, entonces corregimos este problema con un contador que a medida que se van tomando fotos agrega al lado del nombre del archivo un numero que representa la cantidad de veces que se repite el nombre de la imagen al momento de hacer la captura.

```python
def guardar_imagen():
    global Captura
    global ruta_destino
    if Captura is not None and ruta_destino:
        if not os.path.exists(ruta_destino):
            os.makedirs(ruta_destino)

        file_name = entrada_nombre_archivo.get() + ".png"  # Nombre de archivo en entry"
        ruta_guardar = os.path.join(ruta_destino, file_name)

        #Verificar si el archivo ya existe y agregar un número
        contador = 1

        while os.path.exists(ruta_guardar):
            contador += 1
            file_name = entrada_nombre_archivo.get() + f"({contador})" + ".png"
            ruta_guardar = os.path.join(ruta_destino, file_name)

        imagen_pil = Image.fromarray(Captura)
        imagen_pil.save(ruta_guardar)

        print(f"Fotografía guardada como {ruta_guardar}")
```

No te olvides la línea **ventana.mainloop()** al final de nuestro código para que el programa se muestre de forma correcta.

```python
#Botón
BCamara = tk.Button(ventana, text="Iniciar cámara", command=camara)
BCamara.place(x=150,y=330,width=90,height=23)

#Boton para capturar la imagen única
BCapturar = tk.Button(ventana, text="Tomar foto unica", command=Capturar)
BCapturar.place(x=500,y=305,width=100,height=23)

#Boton para capturar la imagen temporizada
BCapturar = tk.Button(ventana, text="Tomar foto temporizada", command=capturar_con_temporizacion)
BCapturar.place(x=500,y=345,width=150,height=23)

# Botón para guardar la imagen
guardar_boton = tk.Button(ventana, text="Guardar Imagen", command=guardar_imagen, state=tk.DISABLED)
guardar_boton.pack(pady=10)
guardar_boton.place(x=580, y=420)

#Cuadro de imagen gris, donde se reproducirá la webcam
LImagen = tk.Label(ventana, background="gray")
LImagen.place(x=50,y=50,width=300,height=240)

#Cuadro de imagen gris, donde se tomara la imagen
LImagenROI = tk.Label(ventana, background="gray")
LImagenROI.place(x=390,y=50,width=300,height=240)

#Boton para seleccionar carpeta
carpeta_seleccionar_button = tk.Button(ventana, text="Seleccionar Carpeta", command=seleccionar_carpeta)
carpeta_seleccionar_button.pack(pady=10)
carpeta_seleccionar_button.place(x = 570, y=380)

ventana.mainloop()
```

Resultado final:



Y al tomar una foto se verá así: