



Universidad Autónoma de San Luis Potosí
Facultad de ingeniería
Tratamiento de Imágenes



Practica 2

Nombre Práctica: Introducción a las
Imágenes Digitales y Video

Nombre del Alumno: Manuel Ramírez Galván

Fecha: 04/02/2025

Procedimiento

2.1.- Diseñe un programa en Python en el cual se cumplan los siguientes requisitos:
Importar la librería de OpenCV.

- Cargar una imagen alojada en el disco duro (escala de grises y a color)
- Primero, la imagen debe de ser mostrada en la ventana creada con el nombre "Ventana 2.1.1" durante 8 segundos.
- Después, la imagen debe de ser mostrada en la ventana creada con el nombre "Ventada 2.1.2" hasta que se presione cualquier tecla.
- Por último, la imagen debe de ser mostrada en la ventana creada con el nombre "Ventada 2.1.3" hasta que se presione la tecla q.

Resultados

```
1  #Ejercicio 1
2
3  import cv2
4
5  img = cv2.imread("C:/Users/HUAMEI/Desktop/L TRATAMIENTO/Manual/Practica 2 - Introduccion a las imagenes digitales y video/Recursos/New_Z
6  img2 = cv2.imread("C:/Users/HUAMEI/Desktop/L TRATAMIENTO/Manual/Practica 2 - Introduccion a las imagenes digitales y video/Recursos/New_Z
7
8  cv2.imshow("Ventana 2.1.1", img)
9  cv2.waitKey(8000)
10 cv2.destroyAllWindows()
11
12 cv2.imshow("Ventana 2.1.2", img)
13 cv2.waitKey(0)
14 cv2.destroyAllWindows()
15
16 mostrar = True
17 while mostrar:
18     cv2.imshow("Ventana 2.1.3", img)
19     keypress = cv2.waitKey(1)
20     if keypress == ord('q'):
21         mostrar = False
22 cv2.destroyAllWindows()
23
24 cv2.imshow("Ventana 2.1.1", img2)
25 cv2.waitKey(8000)
26 cv2.destroyAllWindows()
27
28 cv2.imshow("Ventana 2.1.2", img2)
29 cv2.waitKey(0)
30 cv2.destroyAllWindows()
31
32 mostrar = True
33 while mostrar:
34     cv2.imshow("Ventana 2.1.3", img2)
35     keypress = cv2.waitKey(1)
36     if keypress == ord('q'):
37         mostrar = False
38 cv2.destroyAllWindows()
```

Imagen 1.- Código Ejercicio 1

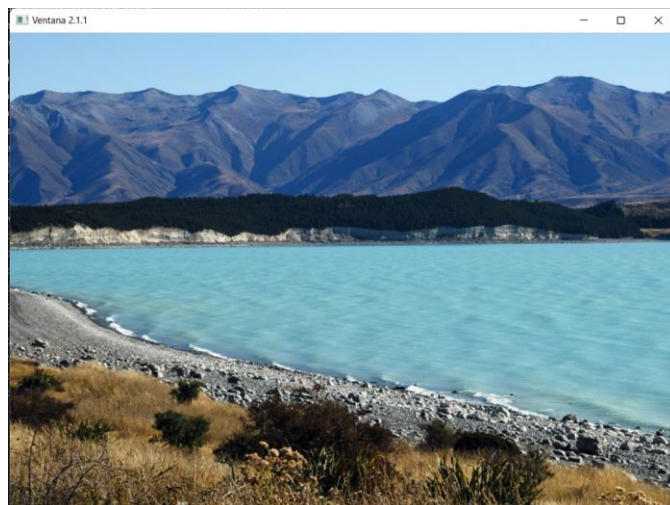


Imagen 2.- Ventana 2.1.1 a Color

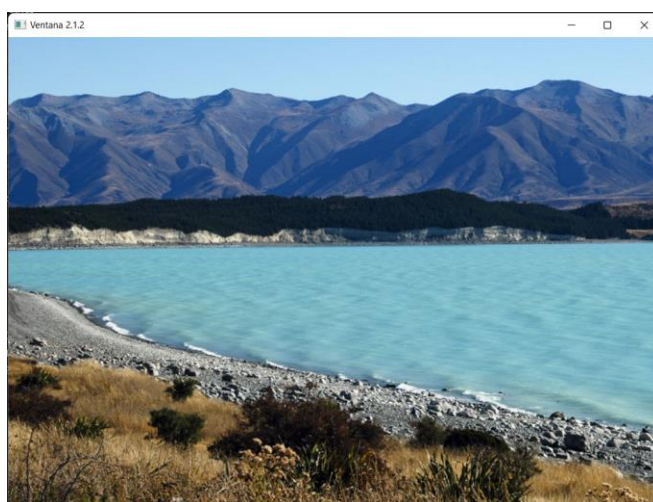


Imagen 3.- Ventana 2.1.2 a Color

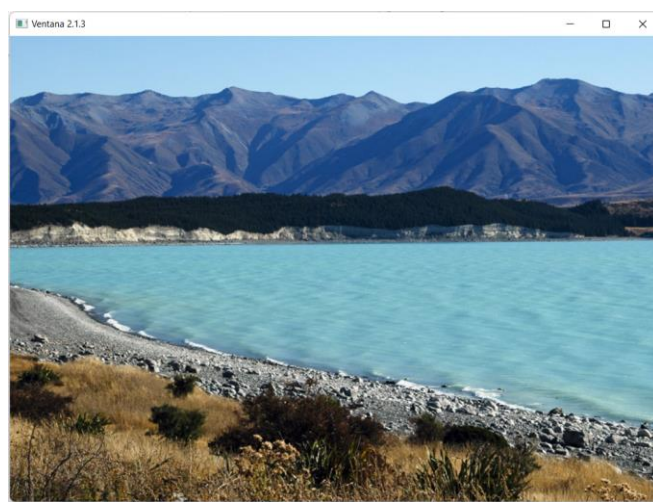


Imagen 4.- Ventana 2.1.3 a Color



Imagen 5.- Ventana 2.1.1 en Escala de Grises



Imagen 6.- Ventana 2.1.2 en Escala de Grises



Imagen 7.- Ventana 2.1.3 en Escala de Grises

Procedimiento

2.2.- Diseñe un programa en Python en el cual se cumplan los siguientes requisitos:
Importar la librería de OpenCV.

- Cargar un video alojado en el disco duro.
- El video debe de ser mostrado en la ventana creada con el nombre "Ejercicio 2.2" hasta que se presione una tecla o se termine el video.

Resultados

```
1 #Ejercicio 2
2
3 import cv2
4
5 vidcap = cv2.VideoCapture("C:/Users/HUAWEI/Desktop/L TRATAMIENTO/Manual/Practica 2 - Introduccion a las imagenes digitales y video/Recu
6 while (vidcap.isOpened()):
7     ret, frame = vidcap.read()
8     if ret:
9         cv2.imshow("Ejercicio 2.2", frame)
10        key = cv2.waitKey(1)
11        if key == ord('q'):
12            break
13    else:
14        break
15
16 vidcap.release()
17 cv2.destroyAllWindows()
```

Imagen 8.- Código Ejercicio 2



Imagen 9.- Ejercicio 2.2

Procedimiento

2.3.- Diseñe un programa en Python en el cual se cumplan los siguientes requisitos:

- Importar la librería de OpenCV.
- Obtenga la captura de video de una cámara web conectada a la computadora (interna o externa).
- La captura de video debe de ser mostrado en la ventana creada con el nombre "Ejercicio 2.3" hasta que se presione la tecla ESC.

Resultados

```

1 #Ejercicio 2
2
3 import cv2
4
5 vidcap = cv2.VideoCapture("C:/Users/HUAWEI/Desktop/L TRATAMIENTO/Manual/Practica 2 - Introduccion a las imagenes digitales y video/Recursos")
6 while (vidcap.isOpened()):
7     ret, frame = vidcap.read()
8     if ret:
9         cv2.imshow("Ejercicio 2.2", frame)
10        key = cv2.waitKey(1)
11        if key == ord('q'):
12            break
13    else:
14        break
15
16 vidcap.release()
17 cv2.destroyAllWindows()

```

Imagen 10.- Código Ejercicio 3

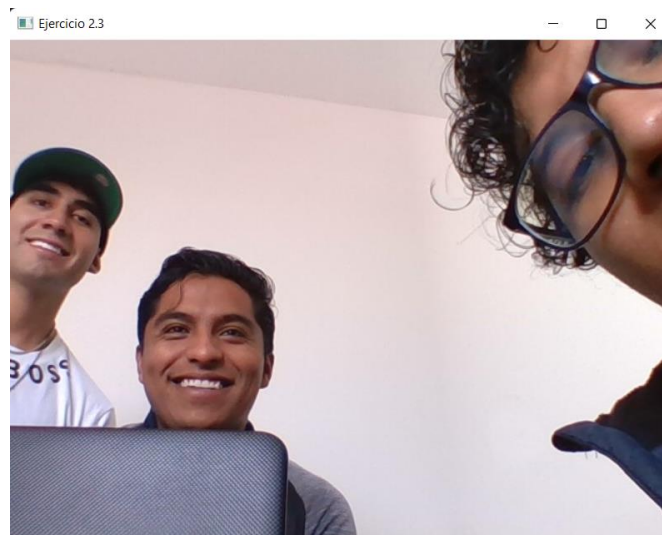


Imagen 11.- Ejercicio 2.3

Comprensión

1. ¿Cuáles son los tipos de imágenes y describa cada una?

Imagen binaria

- Solo puede tener dos valores por píxel (generalmente 0 y 1, o 0 y 255).
- Se utiliza para enmascarar regiones de la imagen, para segmentaciones básicas o para representaciones de alto contraste.

Imagen en escala de grises (grayscale)

- Cada píxel se representa con un solo canal de intensidad (generalmente 8 bits por píxel, valores de 0 a 255).
- Se usan frecuentemente en visión por computadora por ser más sencillas de analizar para muchos algoritmos.

Imagen a color (RGB o BGR)

- Cada píxel tiene tres canales (rojo, verde y azul).
- Puede representarse con 24 bits por píxel (8 bits por canal) u otras profundidades de color.

Imagen con canal alfa (RGBA o BGRA)

- Igual que la imagen a color, pero añade un cuarto canal llamado “alfa”, que se usa generalmente para manejar transparencias.
- Suele representarse con 32 bits por píxel (8 bits por cada canal).

Imágenes con mayor profundidad de bits o en punto flotante

- Se pueden tener imágenes con 16 bits enteros por canal, 32 bits por canal o incluso 64 bits en punto flotante.
- Permiten mayor rango dinámico y precisión en los valores de cada píxel.

2. ¿Para qué se utiliza la función waitKey() en OpenCV?

La función cumple dos propósitos principales:

- Pausa la ejecución del programa durante el número de milisegundos que le especifiquemos como parámetro.
- Captura la pulsación de teclas si ocurre dentro de ese intervalo de tiempo. Devuelve el código de la tecla presionada o -1 si no se presionó ninguna.

Es muy común usar waitKey(0) para pausar la ejecución indefinidamente hasta que el usuario presione una tecla, o usar un pequeño delay en aplicaciones que despliegan un video o realizan un bucle de procesamiento continuo de imágenes.

3. ¿Cuáles son las banderas que se utilizan para recoger los metadatos en la función get()?

Algunos de los más comunes son:

- CAP_PROP_POS_MSEC: Posición en milisegundos del video.
- CAP_PROP_POS_FRAMES: Índice actual de fotograma (frame).
- CAP_PROP_POS_AVI_RATIO: Proporción de progreso en el archivo (0 al inicio, 1 al final).
- CAP_PROP_FRAME_WIDTH: Ancho del frame en píxeles.
- CAP_PROP_FRAME_HEIGHT: Alto del frame en píxeles.

- CAP_PROP_FPS: Cuadros por segundo del video.
- CAP_PROP_FOURCC: Código FOURCC del códec.
- CAP_PROP_FRAME_COUNT: Cantidad total de fotogramas en el video.
- CAP_PROP_BRIGHTNESS: Brillo de la captura (si la fuente lo soporta).
- CAP_PROP_CONTRAST: Contraste de la captura.
- CAP_PROP_SATURATION: Saturación de la captura.
- CAP_PROP_HUE: Tinte (hue) de la captura.
- CAP_PROP_GAIN: Ganancia.
- CAP_PROP_EXPOSURE: Exposición.
- CAP_PROP_CONVERT_RGB: Indica si se convierten los fotogramas a RGB/BGR.
- CAP_PROP_RECTIFICATION: Rectificación (en cámaras estéreo).

Conclusiones

La función `imread()` sirve para leer una imagen, necesita la ruta del archivo y opcionalmente puede tomar la imagen a color, escala de grises, etc.

La función `imshow()` se usa para mostrara una imagen en una ventana de OpenCV, se le puede poner nombre a la ventana; para que la ventana permanezca abierta, se usa la función `waitKey()` y para eliminar la ventana se puede utilizar la función `destroyAllWindows()`.