2025

Reporte

PRACTICA 2

ALEXIS DANIELCORTES TAVERA

Introducción

La visión artificial es un campo de la inteligencia artificial que permite a las computadoras analizar y procesar imágenes digitales.

OpenCV es una de las bibliotecas más utilizadas para estas tareas, ofreciendo herramientas para realizar operaciones aritméticas en imágenes. Estas operaciones incluyen suma, resta, multiplicación y división de imágenes, así como transformaciones como la transpuesta, la rotación y la traslación. En esta práctica, exploramos diversas operaciones sobre imágenes utilizando OpenCV en Python.

Desarrollo Teórico y Practico

En estas prácticas se usaron diversas herramientas para su realización que serán explicadas a continuación:

librerías:

- cv2: Biblioteca OpenCV, usada para el procesamiento de las imágenes.
- numpy: Biblioteca para manejo de arreglos y cálculos numéricos.
- matplotlib.pyplot: Se usa para mostrar imágenes y graficar elementos sobre ellas.

Las operaciones aritméticas en imágenes permiten modificar su apariencia y extraer información relevante.

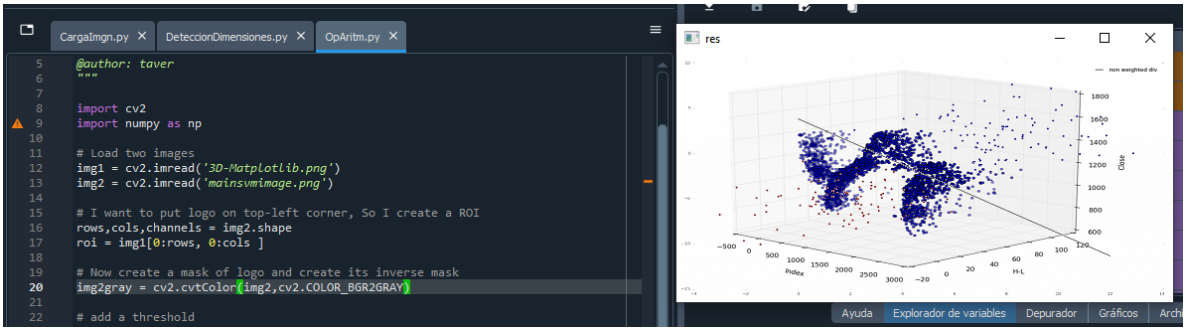
Suma: Permite combinar 2 imágenes aumentando la intensidad de los pixeles

Resta: Se usa para poder resaltar las diferencias entre dos imágenes

Rotación y Traslación: Gira y mueve la imagen respectivamente sin alterarla

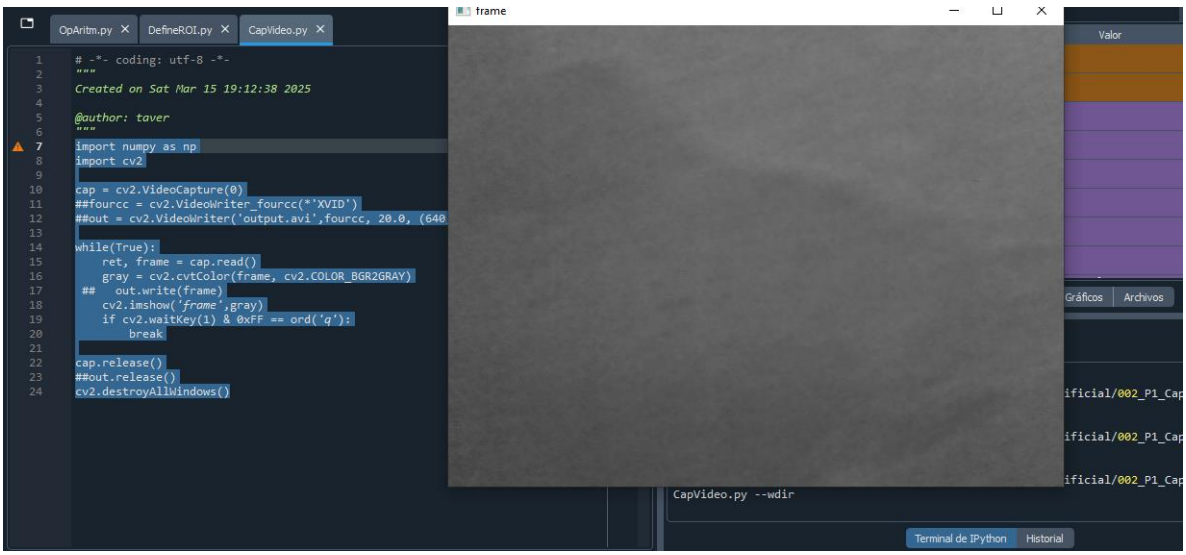
Practica 1:

se utiliza cv2.imread() para cargar imágenes, de las cuales les extraemos una ROI, después se realiza una conversión de color a la segunda imagen que agregamos, una vez esto creamos una mascara binaria para definir el fondo y poder fusionar las imágenes con la función cv2.add(), que sería el equivalente a sumar las imágenes, una vez esto, se mostrará la imagen fusionada,



Practica 2:

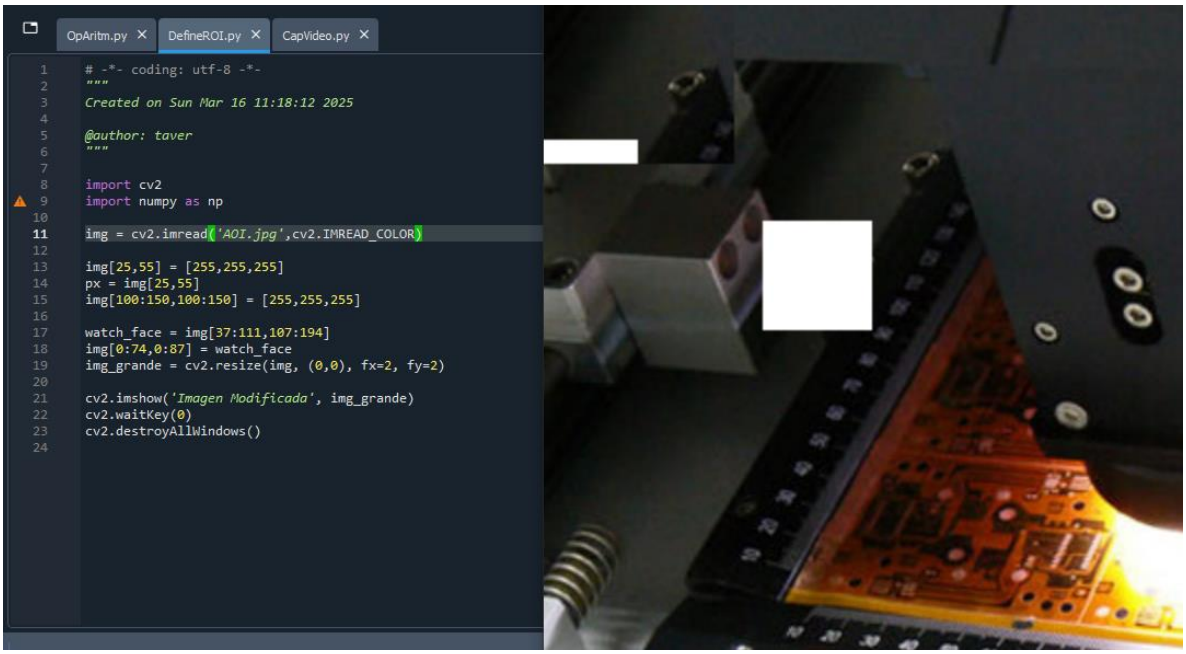
Para esta práctica de procesamiento de video, usaremos una de las funciones de opencv, llamada VideoCapture, la cual controlaremos con un bucle while, debido a que un video no es mas que una secuencia de fotogramas, agregando un filtro blanco y negro



Maestro:
Mauricio Alejandro Cabrera Arellano

Practica 3:

Realizamos la selección de una Región, la cual recolocaremos en una esquina y aumentaremos el tamaño para una mejor visualización



Códigos y funcionamiento:

```
# -*- coding: utf-8 -*-
Created on Sun Mar 16 11:31:30 2025
@author: taver
"""
import cv2
import numpy as np
img1 = cv2.imread('3D-Matplotlib.png')
img2 = cv2.imread('mainsvmimage.png')
rows,cols,channels = img2.shape
roi = img1[0:rows, 0:cols ]
```

Maestro:
Mauricio Alejandro Cabrera Arellano

```
img2gray = cv2.cvtColor(img2,cv2.COLOR_BGR2GRAY)
ret, mask = cv2.threshold(img2gray, 220, 255, cv2.THRESH_BINARY_INV)
mask_inv = cv2.bitwise_not(mask)
img1_bg = cv2.bitwise_and(roi,roi,mask = mask_inv)
img2_fg = cv2.bitwise_and(img2,img2,mask = mask)
dst = cv2.add(img1_bg,img2_fg)
img1[0:rows, 0:cols ] = dst
cv2.imshow('res',img1)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Con el inicio del código se importan las librerías necesarias con los comandos import, para posterior a esto cargar la imagen que queremos visualizar o analizar, en este caso es la imagen de una cámara AOI para inspección de soldadura.

Mostramos la imagen en una ventana y la mantenemos abierta hasta presionar alguna tecla con el comando waitKey(0), junto a DestroyallWindows.

Añadimos la segunda imagen con nuestro filtro para poder visualizarlas juntas en el mismo diagrama

P2

```
# -*- coding: utf-8 -*-
"""
Created on Sat Mar 15 19:12:38 2025
@author: taver
"""
import numpy as np
import cv2
```

Maestro:
Mauricio Alejandro Cabrera Arellano

```
cap = cv2.VideoCapture(0)
##fourcc = cv2.VideoWriter_fourcc(*'XVID')
##out = cv2.VideoWriter('output.avi',fourcc, 20.0, (640,480))
while(True):
    ret, frame = cap.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
 ##   out.write(frame)
    cv2.imshow('frame',gray)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cap.release()
##out.release()
cv2.destroyAllWindows()
```

Con este programa hacemos uso de las diferentes funciones y parámetros para la captura de múltiples imágenes o lo que conocemos como video, todo esto metido en un bucle con orden de finalización al presionar la “q”

P3

```
# -*- coding: utf-8 -*-
"""
Created on Sun Mar 16 11:18:12 2025
@author: taver
"""
import cv2
import numpy as np
img = cv2.imread('AOI.jpg',cv2.IMREAD_COLOR)
```

```
img[25,55] = [255,255,255]
px = img[25,55]
img[100:150,100:150] = [255,255,255]
watch_face = img[37:111,107:194]
img[0:74,0:87] = watch_face
img_grande = cv2.resize(img, (0,0), fx=2, fy=2)
cv2.imshow('Imagen Modificada', img_grande)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Realizamos un recorte y resposicion con una ROI que definimos para poder aumentar la imagen y obtener una mejor visualización

Conclusión

Estas practicas son muy interesantes, realmente mostrando una aplicación muy importante para nuestra actualidad, junto con Python su implementación es bastante sencilla.

Se introducen conceptos esenciales de la visión artificial, como la carga, visualización y manipulación de imágenes.

Las operaciones aritméticas en imágenes permiten realizar transformaciones clave en el procesamiento digital, como combinación de imágenes, enmascaramiento y ajuste de intensidad. Además, la captura de video y la definición de regiones de interés son herramientas fundamentales en aplicaciones de visión por computadora. OpenCV facilita estas operaciones de manera eficiente, permitiendo realizar desde tareas básicas hasta procesamientos avanzados.

Referencias:

Ibm. (2025, 6 enero). Computer Vision. *Vision Artificial*. https://www.ibm.com/mx-es/topics/computer-vision

Maestro:
Mauricio Alejandro Cabrera Arellano