

Reporte

PRACTICA 10

ALEXIS DANIELCORTES TAVERA

Introducción

En esta práctica se trabajó con la extracción de una **región de interés (ROI)** dentro de una imagen y la posterior **detección de esquinas** en dicha región. Esta combinación de técnicas es común en aplicaciones donde se desea enfocar el análisis en una parte específica de una imagen, eliminando el fondo y detectando puntos clave como esquinas o esquemas estructurales.

El enfoque consiste en aislar visualmente la región que nos interesa (por ejemplo, un objeto o una sección de una escena) y luego analizar su estructura mediante un algoritmo de detección de esquinas, como el **método Shi-Tomasi**, que permite identificar características únicas dentro del ROI.

Desarrollo Teórico y Practico

Su objetivo es separar visualmente el fondo de una imagen utilizando una ROI y aplicar detección de esquinas sobre esa región para identificar todos los puntos relevantes de la forma o contorno.

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

librerías:

- cv2: Librería de OpenCV, empleada para cargar la imagen, definir regiones de interés, manipular píxeles y aplicar el algoritmo de detección de esquinas.
- numpy: Utilizada para generar estructuras como máscaras y transformar datos numérico

Maestro:

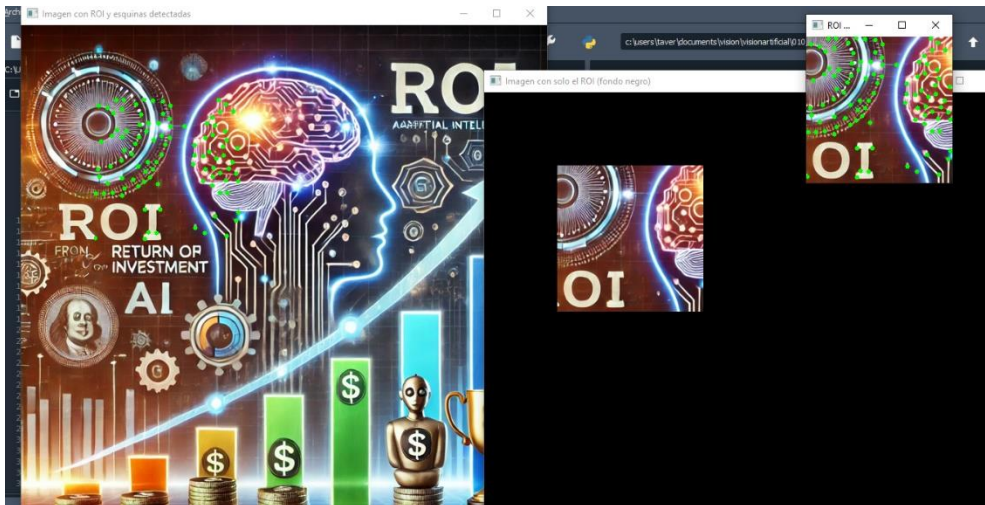
Mauricio Alejandro Cabrera Arellano

Practica 10:

El código realiza dos tareas principales. Primero, extrae una **región de interés (ROI)** específica dentro de la imagen. Para ello, se definen coordenadas rectangulares que indican qué parte de la imagen se desea conservar. Luego, se crea una máscara negra del mismo tamaño que la imagen, y sobre ella se coloca únicamente el contenido de la ROI, lo que permite visualizarla sin el fondo.

Una vez aislada la región, se convierte a **escala de grises**, ya que la detección de esquinas no necesita trabajar con colores, sino con los cambios de intensidad en los píxeles. Luego, se aplica el algoritmo de **Shi-Tomasi** para encontrar las esquinas más relevantes dentro de la ROI. Este algoritmo evalúa qué tan pronunciado es el cambio de intensidad en diferentes direcciones, y selecciona los puntos que presentan las variaciones más fuertes, lo que normalmente corresponde a esquinas.

Finalmente, el código dibuja círculos verdes sobre cada una de las esquinas detectadas y vuelve a insertar la ROI modificada en la imagen original, permitiendo ver claramente los resultados tanto en la región aislada como en el contexto de la imagen completa.



P10

```
import cv2
import numpy as np
img = cv2.imread('imagen.jpg')
img_copy = img.copy()

# Coordenadas de la ROI (x, y, ancho, alto)
```

Maestro:

Mauricio Alejandro Cabrera Arellano

```
x, y, w, h = 100, 100, 200, 200
roi = img[y:y+h, x:x+w]
# Crear máscara negra con solo la ROI
mask = np.zeros_like(img)
mask[y:y+h, x:x+w] = roi
# Convertir ROI a escala de grises
gray_roi = cv2.cvtColor(roi, cv2.COLOR_BGR2GRAY)
# Detectar esquinas en ROI con Shi-Tomasi
corners = cv2.goodFeaturesToTrack(gray_roi, maxCorners=100,
qualityLevel=0.01, minDistance=10)
corners = corners.astype(int)
# Dibujar las esquinas sobre la ROI
for corner in corners:
    cx, cy = corner.ravel()
    cv2.circle(roi, (cx, cy), 3, (0, 255, 0), -1)
# Insertar ROI modificada de vuelta en la imagen
img_with_corners = img.copy()
img_with_corners[y:y+h, x:x+w] = roi

# Mostrar resultados

cv2.imshow('ROI con esquinas', roi)
cv2.imshow('Imagen con solo el ROI (fondo negro)', mask)
cv2.imshow('Imagen con ROI y esquinas detectadas', img_with_corners)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Conclusión

Esta práctica me ayudó a ver con más claridad cómo se puede trabajar con partes específicas de una imagen sin necesidad de procesarla toda. Extraer solo una región de interés (ROI) y ocultar el fondo no solo hace que el análisis sea más rápido, sino también más preciso cuando uno sabe exactamente qué quiere estudiar.

Al aplicar la detección de esquinas dentro de esa región, fue interesante ver cómo el algoritmo localiza puntos clave automáticamente. Esas pequeñas marcas que aparecen pueden parecer simples, pero representan información muy importante que se puede usar para identificar formas, reconocer objetos o seguir movimientos en un video.

Maestro:

Mauricio Alejandro Cabrera Arellano

Lo mejor de todo es que, con muy pocas líneas de código, uno puede lograr algo que tiene muchísimas aplicaciones en la vida real, desde cámaras inteligentes hasta robots. Prácticas como esta hacen que temas que suenan complejos al principio se vuelvan más comprensibles y hasta entretenidos de experimentar.

Referencias:

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

Maestro:

Mauricio Alejandro Cabrera Arellano