
CENTRO DE ENSEÑANZA TÉCNICA INDUSTRIAL
INGENIERÍA EN MECATRÓNICA



Visión artificial

Práctica 4:
ROI

Alumna:
Vanessa Aguirre Diaz

Registro:
22310274

Fecha:
21 de abril del 2025

Desarrollo teórico:

¿Qué es una ROI?

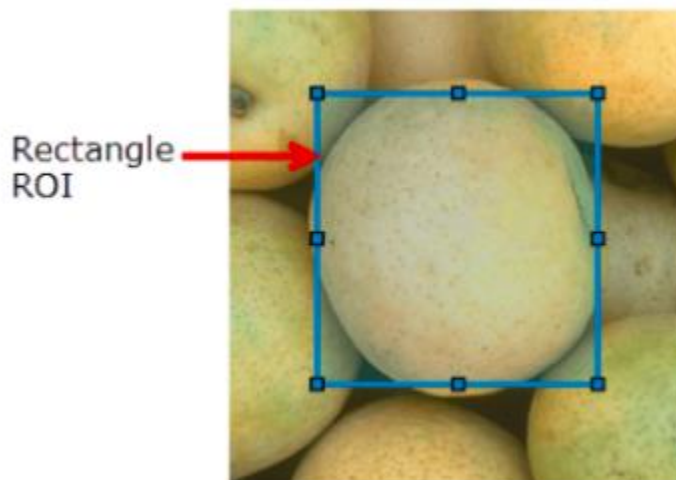
En visión artificial, el término "región de interés" (ROI) designa un área o región específica de una imagen o vídeo que contiene información crucial que debe examinarse y procesarse. La ROI se identifica frecuentemente por su posición, tamaño, forma u otras características visuales, pero también puede ser un subconjunto de la imagen o el vídeo completos.

Las ROI se utilizan en diversas aplicaciones de visión artificial, como el reconocimiento, el seguimiento y la segmentación de objetos. En el seguimiento, una ROI puede describirse como la región por donde se prevé que se desplace el objeto, mientras que en el reconocimiento de objetos, puede definirse como el área que incluye un objeto de interés.

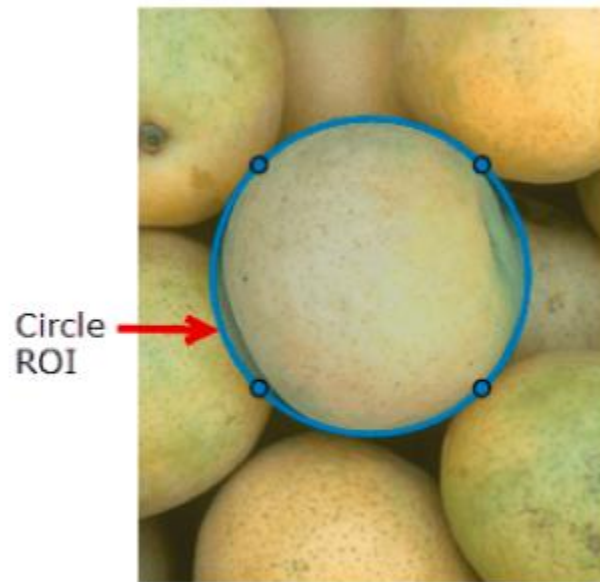
Tipos de ROI

Según las necesidades específicas de la aplicación, las ROI pueden crearse en diversas formas, incluyendo diferentes tamaños, formas y tipos. A continuación, se presentan ejemplos de las ROI más comunes utilizadas en visión artificial:

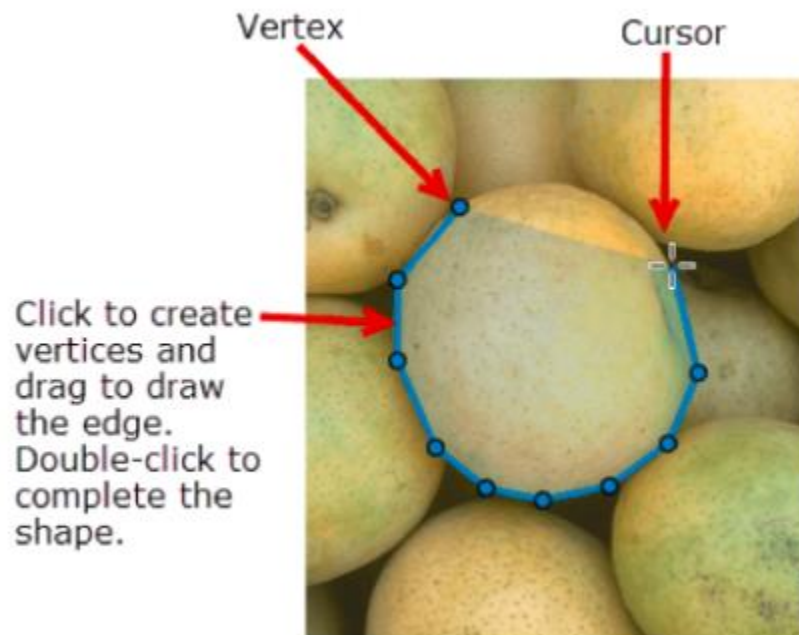
1. **ROI rectangulares:** El ancho y la altura de una ROI rectangular determinan su forma, y su ubicación suele estar determinada por su centro o esquina superior izquierda. Dado que las ROI rectangulares son fáciles de definir e implementar, se utilizan frecuentemente en aplicaciones de visión artificial.



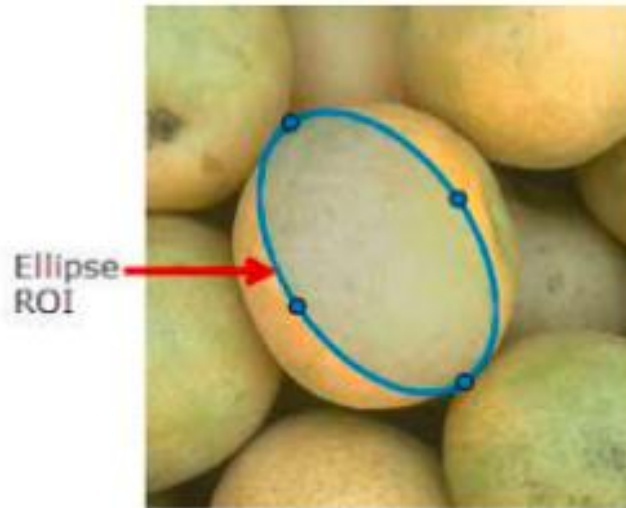
2. **ROI circulares:** Las ROI circulares tienen un radio y un centro, y se emplean con frecuencia en situaciones donde la forma de la ROI es importante. Por ejemplo, ciertas aplicaciones de imágenes médicas emplean ROI circulares para evaluar el tamaño y la forma de los tumores.



- 3. ROI poligonales:** Estas ROI pueden moldearse para adaptarse a los contornos del elemento de interés. Se definen por un grupo de vértices interconectados. Cuando la forma de un objeto es irregular o no uniforme, como en el reconocimiento facial o la identificación de gestos, se utilizan con frecuencia.



- 4. ROI elípticas:** Las ROI elípticas se emplean cuando la forma de la ROI es importante, pero una forma circular no resulta adecuada. Las ROI elípticas se caracterizan por sus ejes mayor y menor, así como por su centro.



¿Por qué se utilizan ROI en visión artificial?

Al definir un ROI, los algoritmos de visión artificial pueden centrar su análisis en un subconjunto de los datos visuales en lugar de procesar la imagen o el vídeo completo.

- El uso de ROI en visión artificial es particularmente significativo en aplicaciones como detección, seguimiento y segmentación de objetos.
- En general, el uso de ROI en la visión artificial permite que los algoritmos centren su análisis en las partes más relevantes e informativas de una imagen o un vídeo, reduciendo el tiempo de procesamiento y mejorando la precisión.

El ROI especifica qué región de la imagen o el vídeo examinará el algoritmo, y elegir el ROI incorrecto puede producir resultados incorrectos o irrelevantes.

- El tamaño del área debe ser una de las principales consideraciones al elegir un ROI.
- Si el ROI es demasiado alto, el algoritmo podría tardar más en procesar los datos y sería más probable que incluyera datos no relacionados en el análisis.
- Por otro lado, si el ROI es demasiado pequeño, podrían pasarse por alto características cruciales, lo que generaría resultados poco confiables.

Selección automática de ROI mediante técnicas como detección de bordes, segmentación de color y detección de objetos

En aplicaciones de visión artificial, la selección automática de ROI mediante métodos como la detección de bordes, la segmentación de color y la detección de objetos es cada vez más común. Estos métodos pueden automatizar la selección de ROI, reduciendo el tiempo empleado y mejorando la precisión.

Al seleccionar el ROI, las técnicas de detección de bordes pueden identificar automáticamente los bordes de los objetos en una imagen o video. Los límites del vehículo pueden utilizarse para definir el ROI en aplicaciones como la detección de vehículos, donde esta estrategia resulta muy útil.

Selección manual de ROI:

- **Identificación de objetos:** Mediante métodos de detección de objetos, la ROI puede seleccionarse automáticamente en función del objeto de interés en la imagen o el vídeo. Esta técnica es rápida y eficaz, lo que la hace adecuada para aplicaciones en tiempo real como la vigilancia o los vehículos autónomos. Sin embargo, requiere un modelo entrenado y podría no ser adecuada para escenas saturadas o complejas.
- **Sustracción de fondo:** Los algoritmos de sustracción de fondo permiten distinguir entre las áreas de primer plano y de fondo en una imagen o vídeo y seleccionar el ROI en función del área de primer plano. La precisión de esta técnica puede aumentar al combinarla con algoritmos de detección de objetos para rastrear objetos en movimiento en vídeos. Sin embargo, podría no ser adecuada en situaciones donde el objeto objetivo esté estático o parcialmente oculto.

Ventajas:

1. El ROI se puede personalizar para satisfacer los requisitos particulares de la aplicación.
2. Se puede utilizar en circunstancias en las que otros métodos podrían no funcionar.
3. Si el usuario está familiarizado con el tema de interés, puede ser más preciso.

Desventajas:

1. Requiere mucho tiempo, en particular si el usuario desea elegir varios ROI si el elemento de interés se mueve rápidamente.
2. Puede ser subjetivo y provocar variaciones en la selección de ROI entre los usuarios.
3. Para aplicaciones en tiempo real, ineficaz.

Selección automática de ROI mediante detección de bordes:

Un enfoque utilizado en visión artificial y procesamiento de imágenes para identificar y extraer regiones de interés en una imagen es la selección automática de ROI (Regiones de Interés) mediante detección de bordes. La detección de bordes es un proceso que analiza los cambios en la intensidad de los píxeles para identificar los límites o bordes de los objetos en una imagen. Estos datos pueden utilizarse para identificar secciones de una imagen que contengan elementos o características de interés.

Ventajas:

1. Es eficaz y rápido, posibilitando aplicaciones en tiempo real.

2. En ocasiones puede ser más preciso que la selección manual, especialmente si el objeto de interés tiene bordes claramente definidos.
3. Se puede utilizar para detectar varios objetos simultáneamente.

Desventajas:

1. Cuando el elemento de interés carece de límites claramente definidos, no puede ser efectivo.
2. Las situaciones con fondos o oclusiones intrincados podrían no funcionar con eficacia.

Desarrollo práctico

En esta practica se realizaron 2 tipos de ROI en Python: una rectangular y otra circular.
Vamos primero con la rectangular:

```
# Practica 4:Extraer una ROI rectangular (Region de interes)

#Importar libreria
import cv2

# Cargar la imagen desde archivo
imagen = cv2.imread('v2.jpeg')

# Coordenadas de la esquina superior izquierda de la ROI
x = 100 # posición en x
y = 150 # posición en y

# Tamaño de la ROI (ancho y alto)
w = 200 # ancho de la ROI
h = 250 # alto de la ROI

# Dibujar un rectángulo verde sobre la imagen para mostrar la ROI
cv2.rectangle(imagen, (x, y), (x + w, y + h), (0, 255, 0), 2)

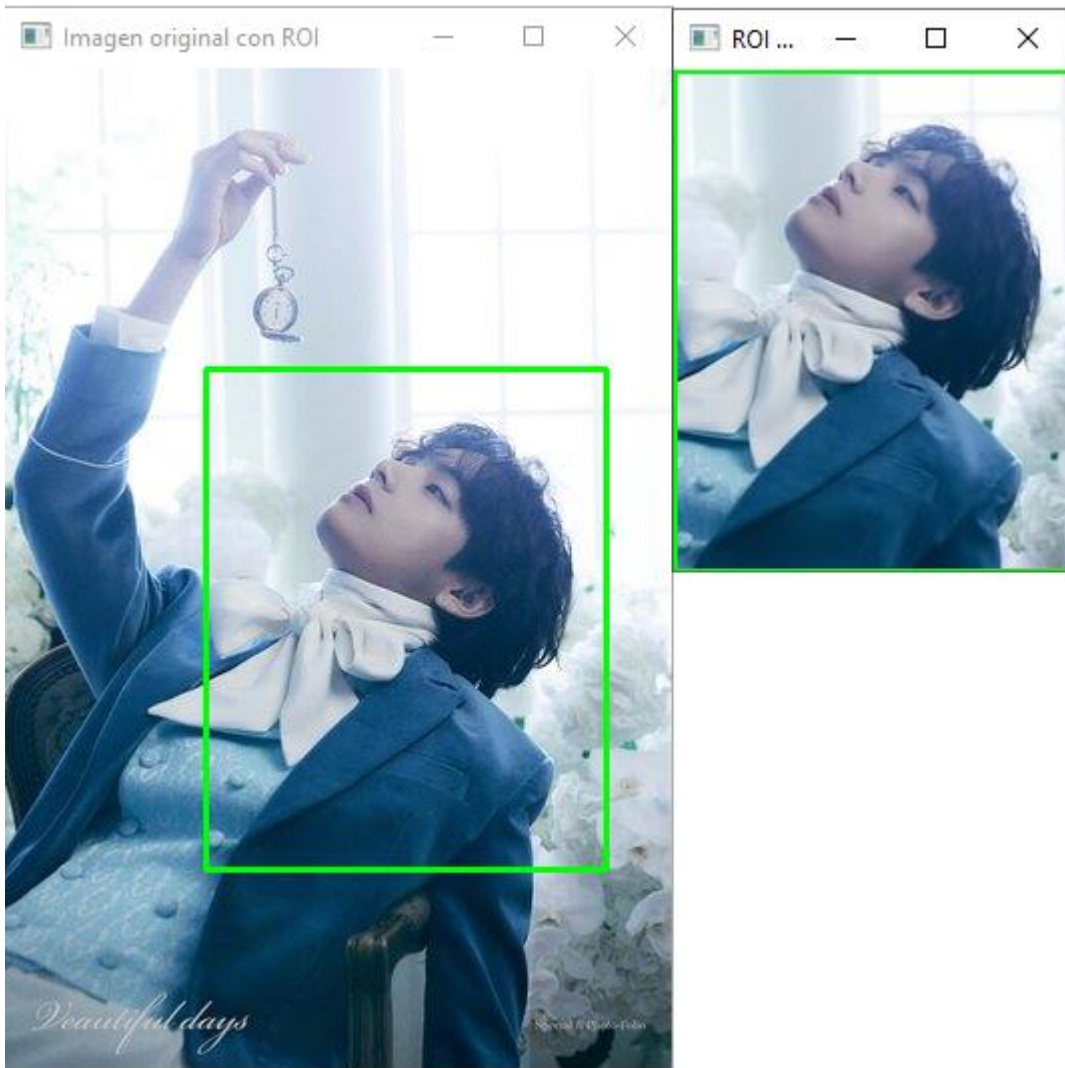
# Extraer la ROI de la imagen (recordar que la imagen se indexa como [fila, columna])
roi = imagen[y:y + h, x:x + w]

# Mostrar la imagen original con el rectángulo de la ROI
cv2.imshow("Imagen original con ROI", imagen)

# Mostrar la ROI extraída por separado
cv2.imshow("ROI extraída", roi)

# Esperar a que el usuario presione una tecla
cv2.waitKey(0)
```

```
# Cerrar todas las ventanas abiertas  
cv2.destroyAllWindows()
```



Ahora vamos con el código para la ROI circular:

```
# Practica 4:Extraer una ROI (Region de interes) circular  
  
#Importar libreria  
import cv2  
import numpy as np  
  
# Cargar la imagen  
imagen = cv2.imread('v2.jpeg')  
  
# Centro del círculo y radio  
centro_x = 200  
centro_y = 250
```



```
radio = 100
```

```
# Dibujar el círculo sobre la imagen original
```

```
cv2.circle(imagen, (centro_x, centro_y), radio, (0, 255, 0), 2)
```

```
# Crear una máscara del mismo tamaño que la imagen, inicialmente negra (cero)
```

```
mascara = np.zeros(imagen.shape[:2], dtype=np.uint8)
```

```
# Dibujar un círculo blanco en la máscara en la posición deseada
```

```
cv2.circle(mascara, (centro_x, centro_y), radio, 255, -1)
```

```
# Aplicar la máscara sobre la imagen para extraer solo la región circular
```

```
roi_circular = cv2.bitwise_and(imagen, imagen, mask=mascara)
```

```
# Mostrar la imagen original con el círculo dibujado
```

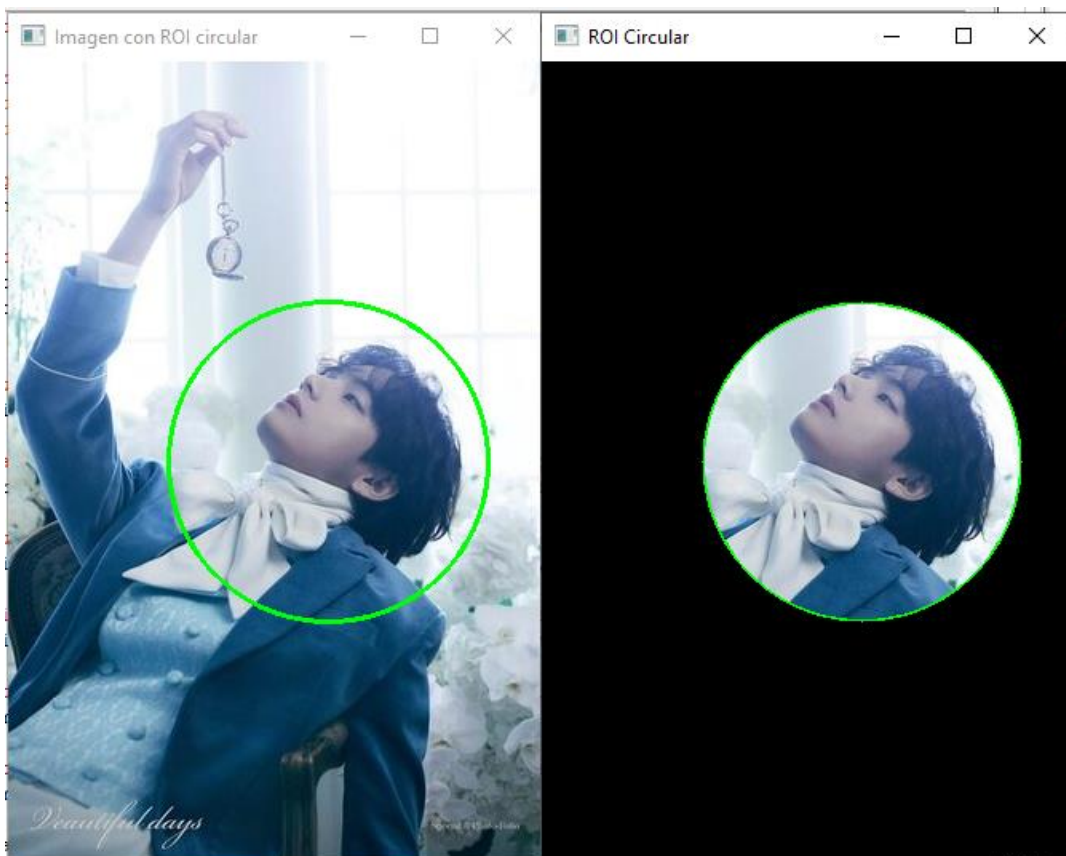
```
cv2.imshow("Imagen con ROI circular", imagen)
```

```
# Mostrar solo la ROI circular (el resto está negro)
```

```
cv2.imshow("ROI Circular", roi_circular)
```

```
cv2.waitKey(0)
```

```
cv2.destroyAllWindows()
```



Para este código se hizo uso de una máscara binaria para aislar la región de interés del resto de la imagen. Esta es una forma fácil de extraer una ROI si esta no es rectangular. Una máscara es una imagen en escala de grises (generalmente en blanco y negro) que actúa como filtro o plantilla para seleccionar partes específicas de otra imagen.

- **Píxeles blancos (valor 255)** → se consideran **válidos** o **visibles**.
- **Píxeles negros (valor 0)** → se **ignoran** o se hacen **transparentes/negros**.

Así, al aplicar una máscara sobre una imagen, estás diciendo: "Solo muéstrame los píxeles donde la máscara es blanca."

Por ultimo tenemos un código que nos permite seleccionar manualmente la ROI con el mouse:

```
#Practica 4: Selecccion manual de una ROI
import cv2

# Cargar imagen
imagen = cv2.imread('v2.jpeg')

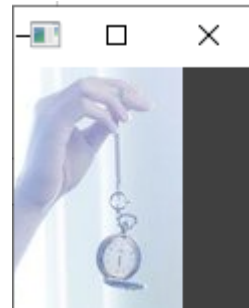
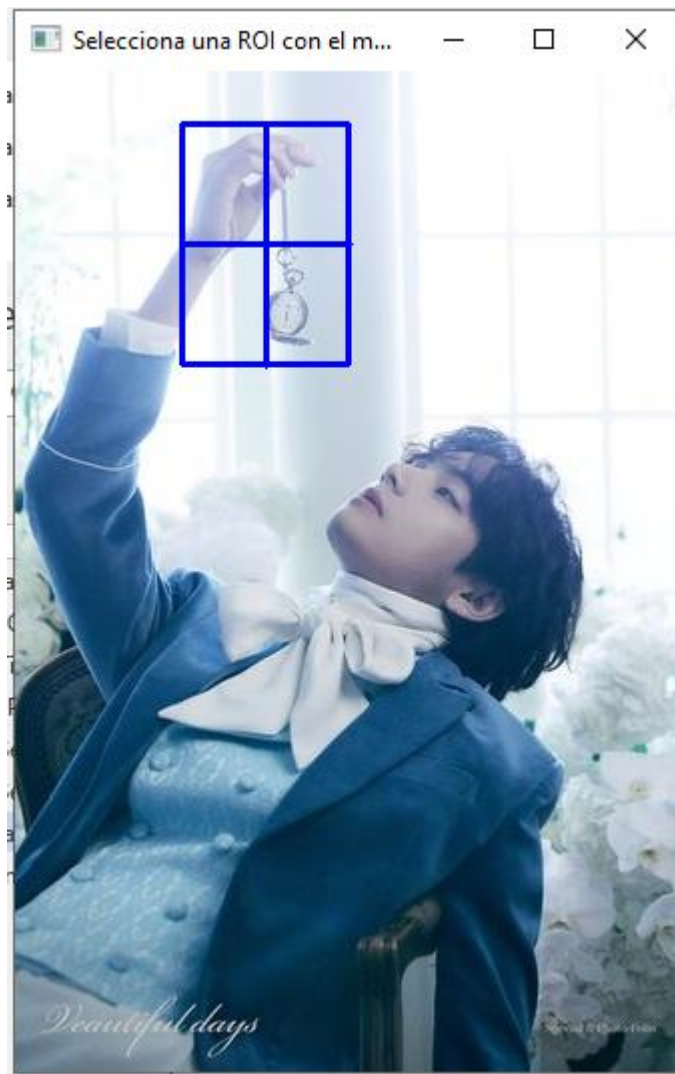
# --- Paso 1: Mostrar ventana para seleccionar ROI manualmente ---
# selectROI devuelve (x, y, w, h)
roi = cv2.selectROI("Selecciona una ROI con el mouse", imagen, fromCenter=False,
showCrosshair=True)

# Destruir la ventana de selección después de usarla
cv2.destroyWindow("Selecciona una ROI con el mouse")

# --- Paso 2: Extraer la ROI usando slicing ---
x, y, w, h = roi
roi_extraida = imagen[y:y+h, x:x+w]

# --- Paso 3: Mostrar la imagen original y la ROI por separado ---
cv2.imshow("Imagen original", imagen)
cv2.imshow("ROI seleccionada", roi_extraida)

# Esperar hasta que se presione una tecla
cv2.waitKey(0)
cv2.destroyAllWindows()
```



Fuente:

<https://www.scaler.com/topics/region-of-interest-opencv/>