



PRACTICA 3.3

Reconocimiento de objetos/prototipos por correlación cruzada.

UPIIT-IPN

25 de junio 2024

Visión Artificial

Italy Abril Zayas Riojas

Homero Meneses Vázquez

Reconocimiento de objetos/prototipos por correlación cruzada.

Introducción: Reconocimiento de Objetos por Correlación Cruzada

El reconocimiento de objetos es una tarea fundamental en el ámbito de la visión artificial y el procesamiento de imágenes. Su objetivo es identificar y localizar objetos específicos dentro de una escena o imagen digital. Entre los diversos métodos existentes para el reconocimiento de objetos, la correlación cruzada se destaca como una técnica simple pero efectiva para encontrar coincidencias entre patrones o prototipos.

En esta práctica, se abordará el desarrollo y aplicación del método de reconocimiento de objetos por correlación cruzada utilizando el lenguaje Python. Se implementará un programa que permita identificar y localizar objetos específicos en una imagen objetivo mediante la comparación con prototipos o plantillas predefinidas.

El método de correlación cruzada se basa en la comparación de la intensidad de píxeles entre una imagen objetivo y un prototipo desplazado en diferentes posiciones. La correlación cruzada produce un mapa de valores que indican la similitud entre la imagen objetivo y el prototipo en cada posición. Los picos de mayor valor en este mapa corresponden a las ubicaciones más probables del objeto dentro de la imagen.

Métodos de Correlación Cruzada

Existen diferentes métodos de correlación cruzada para el reconocimiento de objetos. Los métodos más comunes incluyen:

- **Correlación cruzada espacial:** Este método compara directamente los valores de intensidad de píxeles de la plantilla y la imagen de entrada.
- **Correlación cruzada de fase:** Este método considera la fase de los valores de intensidad de píxeles, lo que puede ser útil para reconocer objetos en imágenes con iluminación variable.
- **Correlación cruzada normalizada:** Este método normaliza los valores de intensidad de píxeles de la plantilla y la imagen de entrada antes de calcular la correlación, lo que puede mejorar la precisión del reconocimiento.

4. Aplicaciones

El reconocimiento de objetos por correlación cruzada se utiliza en una amplia variedad de aplicaciones, incluyendo:

- **Detección de objetos:** Identificar y localizar objetos en imágenes y videos.
- **Seguimiento de objetos:** Rastrear objetos a medida que se mueven en una escena.

- **Clasificación de objetos:** Clasificar objetos en categorías predefinidas.
- **Inspección visual:** Detectar defectos en productos manufacturados.

La práctica se centrará en los siguientes pasos:

1. **Implementación del algoritmo de correlación cruzada:** Se escribirá una función en Python que calcule la correlación cruzada entre dos imágenes.
2. **Carga de imágenes:** Se cargarán la imagen objetivo y los prototipos a utilizar en el proceso de reconocimiento.
3. **Aplicación del algoritmo:** Se aplicará la función de correlación cruzada a la imagen objetivo y cada prototipo.
4. **Localización de objetos:** Se identificarán los picos de mayor valor en los mapas de correlación cruzada para cada prototipo, indicando las ubicaciones probables de los objetos.
5. **Visualización de resultados:** Se mostrarán las imágenes objetivo, los prototipos y los resultados de la correlación cruzada para ilustrar el proceso de reconocimiento.

Esta práctica permitirá comprender los fundamentos del método de correlación cruzada para el reconocimiento de objetos y desarrollar habilidades prácticas en la implementación de algoritmos de visión artificial en Python.

Explicación del Código

Este código en Python realiza la identificación de objetos en una imagen utilizando plantillas (prototipos) y técnicas de procesamiento de imágenes con OpenCV.

Funciones de Procesamiento de Imágenes

load_images(image_path, prototype_path)

Carga la imagen principal y el prototipo desde las rutas especificadas. La imagen principal se carga en color y el prototipo en escala de grises.

process_image(image, prototype, threshold)

Convierte la imagen principal a escala de grises y aplica la técnica de correlación cruzada normalizada para encontrar coincidencias del prototipo en la imagen. Devuelve las coordenadas donde la correlación supera el umbral especificado.

draw_rectangles(image_result, loc, prototype_shape, color)

Dibuja rectángulos en las posiciones encontradas por la correlación cruzada, utilizando el color especificado para cada categoría.

```
save_and_show_image(image_result, output_path)
```

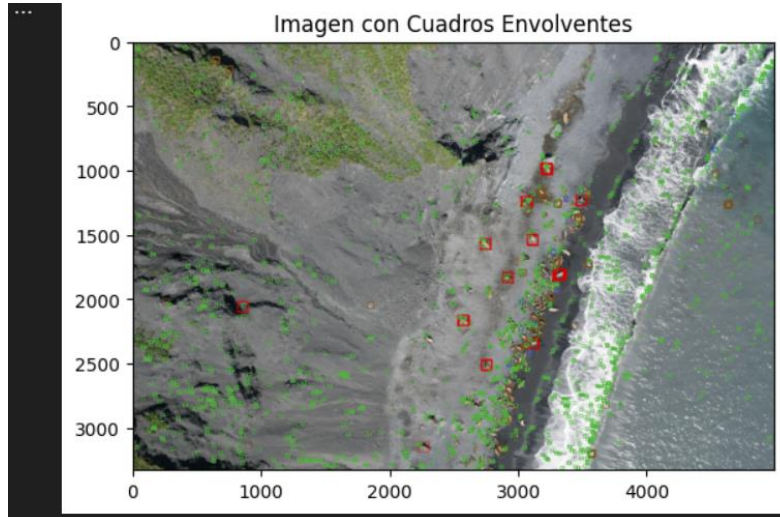
Guarda la imagen con los rectángulos dibujados y la muestra utilizando matplotlib.

```
object_identification(image_path, prototype_path, image_result, color, threshold,  
output_path='image_resultado.jpg')
```

Llama a las funciones anteriores para cargar las imágenes, procesarlas, dibujar los rectángulos y guardar la imagen resultante.

- La función principal: Define las rutas de la imagen principal y de los prototipos, el umbral de correlación y la ruta para guardar la imagen resultante.
- Inicializa `image_result` con la imagen original.
- Organiza los archivos de prototipos en un diccionario según sus categorías (colores).
- Procesa cada prototipo, identificando los objetos en la imagen principal y dibujando los cuadros envolventes.

Este código es un ejemplo práctico de cómo usar OpenCV para la detección de objetos mediante la técnica de correlación cruzada y cómo organizar y visualizar los resultados de manera efectiva.



Conclusión

El reconocimiento de objetos es una tarea fundamental en el ámbito de la visión artificial y el procesamiento de imágenes, cuyo objetivo es identificar y localizar objetos específicos dentro de una escena o imagen digital. En esta práctica, hemos implementado y aplicado el método de correlación cruzada, una técnica simple pero efectiva, utilizando el lenguaje Python y la biblioteca OpenCV.

La correlación cruzada se basa en comparar la intensidad de píxeles entre una imagen objetivo y un prototipo desplazado en diferentes posiciones, produciendo un mapa de valores que indica la similitud en cada posición. Este método, en sus diversas formas (espacial, de fase y normalizada), es eficaz para detectar, rastrear y clasificar objetos en imágenes y videos, así como para inspeccionar visualmente productos manufacturados.

El código desarrollado realiza la identificación de objetos mediante plantillas predefinidas y técnicas de procesamiento de imágenes. Se carga la imagen principal y los prototipos, se aplica la técnica de correlación cruzada normalizada, y se dibujan rectángulos en las posiciones donde se encuentran coincidencias significativas. Además, se organiza y visualiza el resultado de manera efectiva, lo que permite ilustrar el proceso de reconocimiento de objetos.

Este ejercicio no solo proporciona una comprensión sólida de los fundamentos del método de correlación cruzada para el reconocimiento de objetos, sino que también desarrolla habilidades prácticas en la implementación de algoritmos de visión artificial en Python. Así, se demuestra cómo un enfoque sistemático y bien organizado puede facilitar la identificación precisa y eficiente de objetos en imágenes, contribuyendo significativamente a diversas aplicaciones en la industria y la investigación.

Bibliografía

Thanh, N. D., Li, W., & Ogunbona, P. (2010). An Improved Template Matching Method for Object Detection. En *Lecture notes in computer science* (pp. 193-202).

https://doi.org/10.1007/978-3-642-12297-2_19