

Reconocimiento de Patrones mediante Template Matching usando OpenCV

José Alvarado Espinoza,
Pontificia Universidad Católica de Valparaíso,

Abstract—En este artículo, exploraremos cómo implementar el template matching en Python utilizando OpenCV, describiendo los conceptos teóricos detrás de esta técnica y demostrando su aplicación práctica en diferentes escenarios de reconocimiento de imágenes. Además, discutiremos las ventajas y limitaciones del template matching y cómo se compara con otros métodos más avanzados de reconocimiento de imágenes.

Index Terms—Template Matching, OpenCV, reconocimiento de imágenes, Python.

I. INTRODUCTION

El reconocimiento de imágenes es una disciplina fundamental dentro del campo de la visión por computadora y se ha convertido en una tecnología esencial en diversas aplicaciones como la seguridad, la medicina, la automatización industrial y la realidad aumentada. Una de las técnicas más sencillas es el "template matching" o "correspondencia de plantillas".

El template matching es un método basado en la búsqueda de una subimagen o plantilla dentro de una imagen más grande. Este enfoque se basa en la correlación, donde se desliza una plantilla sobre la imagen de entrada y se calcula una medida de similitud en cada posición. Los puntos con mayor similitud indican la presencia de la plantilla en la imagen. Esta técnica es particularmente útil para tareas donde se busca un patrón específico en una imagen, como la detección de objetos, reconocimiento de caracteres o seguimiento de movimiento.

II. MARCO TEÓRICO

La correlación cruzada normalizada es una técnica poderosa y sencilla para el reconocimiento de patrones en imágenes. A través de la normalización, se logra una medida robusta que es independiente de las variaciones de escala e iluminación, haciendo que sea una herramienta eficaz para diversas aplicaciones en visión por computadora.

A. Conceptos Clave

El template matching es una técnica fundamental en el reconocimiento de patrones dentro del campo de la visión por computadora. Se utiliza para localizar una subimagen o plantilla T dentro de una imagen más grande I . Una de las formas más robustas de realizar template matching es mediante la *Correlación Cruzada Normalizada* (NCC, por sus siglas en inglés). A continuación, se presenta la fórmula matemática para la NCC:

$$R(x, y) = \frac{\sum_{x', y'} (T'(x', y') \cdot I'(x + x', y + y'))}{\sqrt{\sum_{x', y'} (T'(x', y')^2) \cdot \sum_{x', y'} (I'(x + x', y + y')^2)}}$$

Donde:

- T es la imagen de la plantilla.
- I es la imagen en la que se busca la plantilla.
- (x, y) son las coordenadas en la imagen I donde se calcula la similitud con la plantilla.
- (x', y') son las coordenadas dentro de la plantilla T .
- $T'(x', y')$ representa los valores de los píxeles de la plantilla centrados en el punto medio de T .
- $I'(x + x', y + y')$ representa los valores de los píxeles de la imagen en la ventana definida por la plantilla centrada en el punto (x, y) .

B. Explicación de la Fórmula

1) Numerador (Producto punto):

$$\sum_{x', y'} (T'(x', y') \cdot I'(x + x', y + y'))$$

Este término calcula la suma del producto de los valores de píxeles correspondientes de la plantilla y la imagen. Este producto punto mide la similitud directa entre la plantilla y la región correspondiente en la imagen.

2) Denominador (Normalización):

$$\sqrt{\sum_{x', y'} (T'(x', y')^2) \cdot \sum_{x', y'} (I'(x + x', y + y')^2)}$$

Este término normaliza el producto punto dividiéndolo por la raíz cuadrada de la suma de los cuadrados de los píxeles en la plantilla y en la región de la imagen. La normalización es crucial para asegurar que la medida de similitud no dependa de la escala de intensidad de los píxeles. Esto permite que la correlación sea robusta a cambios en la iluminación y contraste.

C. Interpretación del Resultado

Los valores de $R(x, y)$ varían en el rango de $[-1, 1]$:

- Un valor cercano a 1 indica una alta similitud entre la plantilla y la región de la imagen, sugiriendo una posible coincidencia.
- Un valor cercano a 0 indica baja similitud.
- Un valor cercano a -1 indica una fuerte anticorrelación, lo que generalmente no es útil para la detección de la plantilla.

D. Aplicación Práctica

En la práctica, esta fórmula se aplica en cada posición (x, y) dentro de la imagen I para calcular $R(x, y)$. Se busca la posición donde $R(x, y)$ es máxima, lo que indica la mejor coincidencia de la plantilla dentro de la imagen.

III. METODOLOGIA

A. Obtención de Imágenes de Referencia

Se obtuvieron tres imágenes de referencia que se utilizarán para el procesamiento y comparación:

- `box_ref.png`: Imagen de referencia de una caja.
- `circ_ref.png`: Imagen de referencia de un círculo.
- `cross_ref.png`: Imagen de referencia de una cruz.

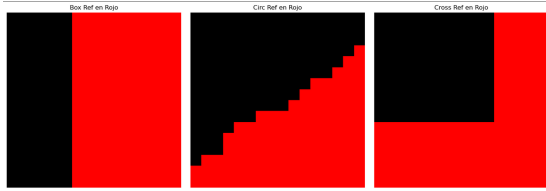


Fig. 1: Imágenes que usaremos de kernel.

B. Separación y Procesamiento de Imágenes de Entrada

Las imágenes de entrada se procesaron y se separaron en carpetas específicas según su tipo (caja, círculo, cruz):

- Las imágenes se leyeron desde la carpeta `processed512`.
- Cada imagen se procesó para conservar solo el canal rojo.
- Se separaron en subcarpetas (`box`, `circ`, `cross`) dentro de la carpeta `Imagenes_separadas`.

C. Recorte de Imágenes

Las imágenes separadas se recortaron a un tamaño de 256x256 píxeles (porción superior izquierda):

- Las imágenes de tamaño 512x512 píxeles se recortaron y almacenaron en una nueva carpeta llamada `separadas_cortadas`.

D. Template Matching

Se realizó la técnica de *Template Matching* para comparar las imágenes recortadas con las plantillas generadas:

- Las plantillas se leyeron desde la carpeta `plantillas`.
- Se utilizó la función `cv2.matchTemplate` con el método `cv2.TM_SQDIFF` para encontrar la mejor coincidencia.
- La imagen de entrada se clasificó y renombró según la mejor coincidencia encontrada.

E. Resultados de Clasificación

Se contabilizaron los verdaderos positivos y los falsos positivos para evaluar la precisión del método.

IV. RESULTADOS

A. Clasificación

Se procesaron y clasificaron las imágenes utilizando la técnica de Template Matching. A continuación, se presentan los resultados de la clasificación para cada tipo de imagen.

TABLE I: Resultados de Clasificación

Tipo	Verdaderos Positivos	Falsos Positivos
Total	7499	6829

B. Tiempo de Procesamiento

El tiempo de procesamiento para la clasificación de las imágenes fue de entre 5 a 8 minutos, dependiendo de la capacidad de la máquina utilizada. Este tiempo incluye la lectura y clasificación de las imágenes.

C. Ejemplos Visuales

A continuación, se muestran algunos ejemplos de verdaderos positivos y falsos positivos para cada tipo de imagen.

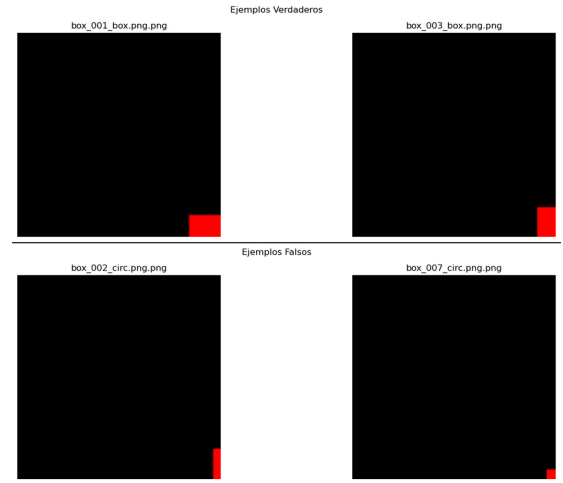


Fig. 2: Ejemplos de Verdaderos Positivos y Falsos Positivos.

V. CONCLUSIONES

Los resultados indican un rendimiento del 50% en la clasificación, con un número significativo de falsos positivos. El método de Template Matching, aunque sencillo y eficiente en términos de tiempo, presenta limitaciones en precisión que deben ser abordadas en trabajos futuros mediante técnicas más avanzadas de reconocimiento de imágenes.

El alto número de falsos positivos sugiere que la técnica puede no ser suficientemente robusta frente a variaciones en las imágenes, como cambios en la iluminación, escala y rotación. Esto plantea la necesidad de explorar métodos más avanzados para mejorar la precisión del reconocimiento de patrones.

A. Trabajo Futuro

Para futuros trabajos, se recomienda explorar técnicas de Deep Learning, como redes neuronales convolucionales (CNNs), que han demostrado un rendimiento superior en tareas de reconocimiento de imágenes. Las CNNs pueden aprender características más complejas y abstractas de las imágenes, lo que potencialmente reducirá la tasa de falsos positivos y aumentará la precisión general del sistema.

La implementación de técnicas de Deep Learning requerirá un mayor poder computacional y una cantidad considerable de datos etiquetados para el entrenamiento. Sin embargo, los beneficios en términos de precisión y robustez justifican la inversión. Además, se pueden considerar enfoques híbridos que combinen Template Matching con aprendizaje profundo para aprovechar las fortalezas de ambos métodos.

En resumen, mientras que el Template Matching proporciona una solución inicial viable y rápida, la incorporación de Deep Learning en futuros estudios promete mejoras significativas en la precisión y fiabilidad del reconocimiento de patrones en imágenes.

REFERENCES

- [1] Szeliski, R. (2010). *Computer Vision: Algorithms and Applications*. Springer.
- [2] Gonzalez, R. C., & Woods, R. E. (2007). *Digital Image Processing*. Pearson Prentice Hall.
- [3] OpenCV Documentation. (n.d.). Template Matching. Recuperado de https://docs.opencv.org/4.x/d4/dc6/tutorial_py_template_matching.html