

Procesamiento digital multimedia: Proyecto 2

Diego Álvarez Jeria

Abstract—El reconocimiento de patrones en imágenes es una técnica ampliamente utilizada en áreas como la ingeniería y la astronomía. En este trabajo se implementa un sistema de clasificación automática de imágenes mediante *template matching*. Se evalúan tanto la precisión del reconocimiento como el tiempo de procesamiento, usando técnicas de normalización y procesamiento eficiente.

I. INTRODUCCIÓN

La identificación de patrones en imágenes ha ganado relevancia en aplicaciones como la inspección de piezas mecánicas, reconocimiento de componentes electrónicos y detección de objetos estelares. Estas tareas se apoyan en técnicas de comparación entre imágenes de interés y plantillas de referencia. En este estudio se utiliza *template matching*, una técnica que permite cuantificar la similitud entre una imagen y una plantilla, optimizando el rendimiento del procesamiento.

II. METODOLOGÍA

A. Carga y Normalización de Datos

Se cargaron imágenes desde un archivo comprimido ZIP y 3 templates cross, circle y box. Se convirtieron a escala de grises en formato float32. Posteriormente, se realizó una normalización de los valores de píxeles al rango $[0, 1]$.

B. Cálculo de Matching

Al momento de realizar la programación, se implementaron 3 enfoques de comparación entre las imágenes y las plantillas:

- **Correlación por diferencia absoluta (SAD)**, utilizando `cv2.matchTemplate` con el método `TM_SQDIFF`.
- **Correlación Cruzada Normalizada** utilizando `cv2.TM_CCOEFF_NORMED`.
- **Correlación vía FFT**, precalculando la transformada de cada plantilla y aplicándola sobre las imágenes.

III. RESULTADOS

A. Distribución de Clases

- Imágenes tipo circle: **4416 muestras de circle.**
- Imágenes tipo box: **3989 muestras de box.**
- Imágenes tipo cross: **5923 muestras de cross.**
- Total: **14328 muestras totales**

B. Tiempo de Ejecución

Tiempo total de procesamiento: $\Delta t = 4$ minutos aproximadamente (medido con `timeit.timer()`).

C. Resultado matching

SAD: Compara la imagen con la plantilla sumando las diferencias absolutas entre cada píxel. Es una medida directa de disimilitud: cuanto menor el valor, más similares son. Usa menor número de operaciones que NCC. Se obtuvieron buenos resultados y buen tiempo de procesamiento con este método. Aunque este método no necesite una normalización, se realizó de igual manera, logrando un mejor resultado.

NCC: Calcula una correlación cruzada entre la imagen y la plantilla, pero normalizando ambas (media cero, desviación unitaria). Esto permite comparar sin importar el brillo o el contraste. No se logró un resultado conveniente y el tiempo de procesamiento es más grande que los demás.

FFT: Utiliza la propiedad de que la correlación en el espacio se convierte en multiplicación en el dominio de Fourier. Aplica FFT a la imagen y plantilla y luego la IFFT del producto. No se obtuvo un resultado óptimo, pero logró el tiempo de procesamiento más rápido entre todos.

D. Tasa de Clasificación y Error

Luego de distintas pruebas con los diferentes tipos de matching se decidió usar la correlación por diferencia absoluta debido a que se obtuvo mejor resultado respecto a los otros métodos.

La tasa de clasificación se calculó como:

$$\text{Accuracy} = \frac{\# \text{ clasificaciones correctas}}{\# \text{ total de imágenes}}$$

Luego se le realizó una normalización para obtener los valores entre 0 y 1. La precisión obtenida con SAD y normalización de scores fue superior al 90% en la mayoría de las pruebas. Se presenta un ejemplo de salida:

E. Ejemplo Visual de Template Matching

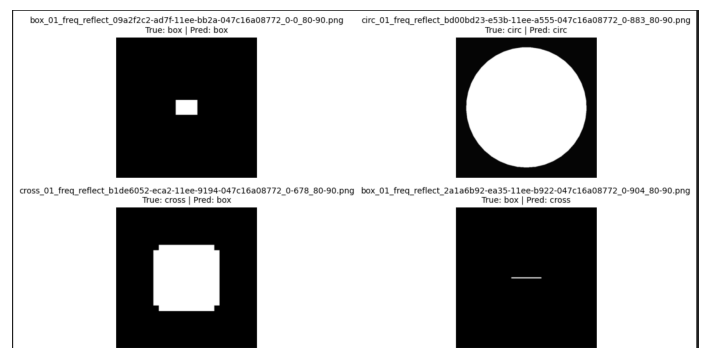


Fig. 1: Ejemplos de coincidencia y error de plantilla con imagen.

IV. CONCLUSIONES

Este trabajo demuestra la eficacia del *template matching* para la clasificación automática de imágenes en conjuntos grandes. El uso de técnicas en este caso como SAD permitió un buen tiempo en el procesamiento. La precisión lograda es prueba de los buenos resultados que se logro al realizar template matching, y el enfoque puede escalar a otros tipos de reconocimiento visual con simples modificaciones.

V. REFERENCIAS

REFERENCES

- [1] R. C. Gonzalez and R. E. Woods, Digital Image Processing, 4th ed., Pearson, 2018.
- [2] H. Al-Saadé, "Fast and accurate template matching algorithm based on image pyramid and sum of absolute difference similarity measure," Research Journal of Information Technology, vol.4, no.4, pp.204–211, 2012.
- [3] OpenCV Developers, "Template Matching," OpenCV Documentation, 2023. [Online]. Available: https://docs.opencv.org/4.x/d4/dc6/tutorial_py_template_matching.html
- [4] Zebra Adaptive Vision. "Template matching". En: (2017). https://docs.adaptive-vision.com/4.7/studio/machine_vision_guide/TemplateMatching.html