

Reconocimiento de imágenes utilizando Template Matching

Simón Delgado, *Estudiante Ingeniería Civil Eléctrica, PUCV,*

Abstract—Este documento presenta el desarrollo de un algoritmo de clasificación de imágenes en el cual, al tener un énfasis en la celeridad del proceso, se emplea preprocesamiento como binarización y redimensionamiento, para luego realizar la predicción de la categoría de la imagen a través de correlaciones entre éstas y los templates, para luego aplicar un proceso de verificación de los resultados con tal de medir el grado de exactitud del algoritmo.

Index Terms—Template Matching, binarización, clasificación de imágenes, preprocesamiento.

I. INTRODUCTION

EL reconocimiento de imágenes es un campo en constante desarrollo gracias a sus múltiples aplicaciones, como seguridad o visión artificial. Si bien existen algoritmos de procesamiento avanzados para cumplir con esta tarea, el presente trabajo se enfoca en la optimización de los recursos de programación para permitir la clasificación de una gran cantidad de imágenes en un corto periodo de tiempo a través de técnicas de preprocesamiento.

II. MARCO TEÓRICO

El Template Matching consiste en el reconocimiento de objetos en imágenes a través del nivel de correlación entre la imagen y una imagen de prueba o "template". Si bien esta técnica puede ser considerada básica, la simpleza de su aplicación permite enfocarse en otros problemas [2] como el preprocesamiento para la optimización del proceso de clasificación de imágenes. Las imágenes digitales son tensores los cuales, normalmente, constan de 3 canales ROJO, AZUL y VERDE. Estos canales son matrices que se superponen para generar la imagen final, en donde cada pixel es una combinación de los 3 colores. La binarización elimina la necesidad de 3 canales y genera una imagen de 1 solo canal, lo que disminuye la cantidad de información a analizar. Bajo esta lógica, el redimensionamiento disminuye la cantidad de saltos que debe dar el template en el proceso de Matching lo que ahorra muchos recursos computacionales.

III. METODOLOGÍA

Para este trabajo, se utilizó un preprocesamiento para las imágenes el cual consistió en una binarización y resizing de éstas y de los templates. Las imágenes pueden pertenecer a las categorías "box", "circ" y "cross" (cuadrados, círculos y cruces), y los templates presentan formas con características de una de estas 3 categorías. Las figuras 1 y 2 presentan la comparación entre una imagen "box" original y una preprocesada, con los ejes indicando la cantidad de píxeles.

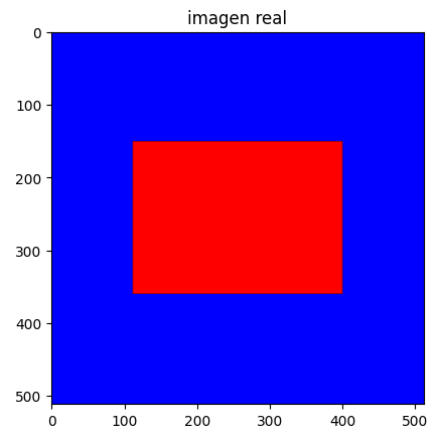


Fig. 1: Original.

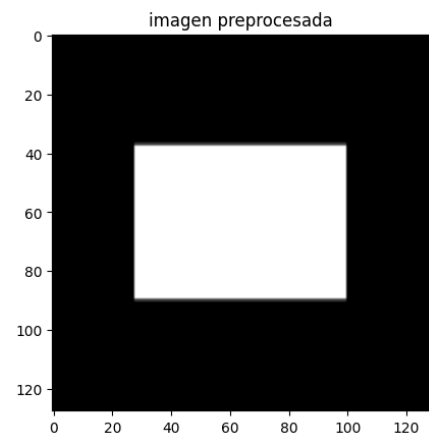


Fig. 2: Preprocesada.

El resizing de de las imágenes se hizo de 512x512 a 128x128 píxeles y el de los templates de 32x32 a 16x16. Éstos tamaños resultaron ser los más óptimos en cuanto a recursos computacionales y precisión de la predicción.

Luego del preprocesamiento, se procedió a hacer el Matching de las imágenes. Este proceso consiste en un escaneo de la imagen preprocesada con el template también preprocesado, el cual arroja un grado de correlación. Para aumentar la capacidad de predicción, se optó por generar 2 procesos de Matching: uno con el template normal y otro con el template rotado en 180°.

Para el proceso de clasificación, se utiliza el grado de correlación más alto de los 6 Matching (3 template con 2 rotaciones) y se clasifica según la categoría del template correspondiente. La clasificación real de las imágenes se obtiene a

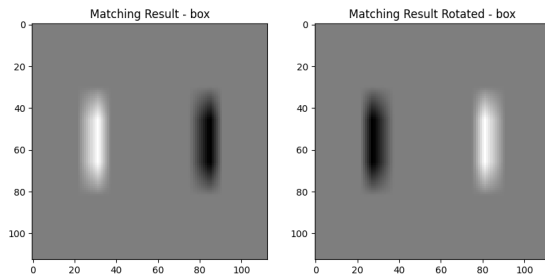


Fig. 3: Matching Box Template

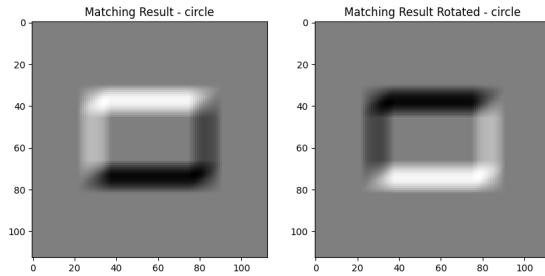


Fig. 4: Matching Circle Template

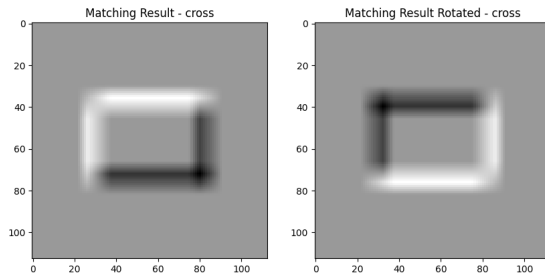


Fig. 5: Matching Cross Template

través de su nombre, ya que llevan la etiqueta "box", "cross" y "circ". Para encontrar el grado de precisión, se hace una comparación porcentual entre las clasificaciones predichas y las reales.

IV. RESULTADOS

Los resultados, naturalmente, fueron variando a lo largo del desarrollo del algoritmo de predicción. El primer algoritmo terminado arrojó cerca de un 37% de precisión. Este primer algoritmo contempló un filtro sobel que, al ser eliminado, se acercó a un 50% de precisión. Luego de variar diferentes parámetros como el tamaño de resizing y el tipo de Matching (siendo ".TM_CCOEFF" el más óptimo [1]) se consiguió un grado de exactitud de 76.80%, 3 de cada 4, y con un tiempo de procesamiento de alrededor de 37 segundos desde el Matching hasta la obtención del grado de error.

V. CONCLUSIONES

Luego de haber obtenido el mayor grado de precisión (76.80%), no fue posible aumentarlo mediante la manipulación de los parámetros ya ingresados. Esto sugiere que, si se quiere aumentar la exactitud del algoritmo de predicción, hará falta agregar más aspectos al preprocesamiento y el proceso de Matching. Por otro lado, teniendo en cuenta el tamaño del

conjunto de datos sobre el que se está trabajando (alrededor de 14000 imágenes) y que el Matching se aplica 6 veces, el tiempo de proceso es sumamente bajo. Quizá se comprometió precisión por tiempo de procesamiento cuando deba ser el caso contrario. Finalmente, el desarrollo del algoritmo de Template Matching puso énfasis en los aspectos de preprocesamiento que permitieran optimizar el proceso, lo cual a través de manipulación de los parámetros y de elección de un preprocesamiento simple pero eficiente, permitió aumentar sustancialmente desde app. 40% a app. 77% sin una variación significativa en el tiempo de procesamiento.

REFERENCES

- [1] *OpenCV: Template Matching*. OpenCV Documentation. [Online]. Available: https://docs.opencv.org/4.x/d4/dc6/tutorial_py_template_matching.html. [Accessed: Jun. 23, 2024].
- [2] R. Brunelli, *Template Matching Techniques in Computer Vision: Theory and Practice*. Wiley, 2009.
- [3] *Template Matching*. Adaptive Vision, [Online]. Available: https://docs.adaptive-vision.com/4.7/studio/machine_vision_guide/TemplateMatching.html. [Accessed: Jun. 23, 2024].
- [4] *UMBRALIZACIÓN DE UNA IMAGEN*. Unipython, [Online]. Available: <https://unipython.com/umbralizacion-una-imagen/>. [Accessed: Jun. 23, 2024].