

# Tarea 1 - Reconocimiento de Patrones

Bastían Marinkovic

Pontificia Universidad Católica de Chile

**Motivación**—Esta tarea busca introducir al estudiante a técnicas usadas en el reconocimiento de patrones, esto mediante la detección de números presentes en imágenes. Este tipo de problema puede ser muy útil en la automatización de la detección de números, algunos ejemplos de sus aplicaciones es la verificación de si un automóvil ha superado el límite de velocidad indicado por un medidor, otra implementación se puede dar en control de calidad de etiquetas numéricas.

## I. SOLUCIÓN PROPUESTA

El algoritmo realiza un preprocesamiento a la imagen antes de realizar la determinación de los números presentes en esta. El preprocesamiento consiste en, primero escalar los colores de la imagen para que queden en un rango entre 0-255, luego se realiza un cambio de color para solo utilizar tonos de gris, sobre dicho resultado se aplica un filtro de los tonos de gris menores a 50 con esto se va a tener una máscara la que será utilizada para etiquetar los números con la función *Label* de *skimage*, gracias a esto se puede individualizar cada número presente en la imagen y realizar la detección de cada uno.

Con dicho resultado se calcula la *bounding box* que cubre al número y se llama a la función *Determine\_number* que es la encargada de retornar el número determinado, para esta operación en primer lugar se umbraliza la imagen y los valores resultantes que se encuentran sobre el umbral se dejan en 255, en otro caso 0. Con estos pasos se procede a realizar un árbol de decisión para la elección de los números.

Este árbol comienza con un caso base para determinar si la *bounding box* es 1, para esto se calcula la cantidad de píxeles pintados en la caja si es mayor a un *threshold* se retorna 1, sino se pasa al siguiente nivel del árbol donde se determina si es que el número presenta agujeros o no, esto se hace mediante el uso de la función *binary\_fill\_holes* y la diferencia en área cubierta por el número, en caso de tener agujeros se presentan 4 casos caracterizados por la posición del agujero en la imagen. El primer caso si solo hay agujeros en la mitad superior de la *bounding box* se retorna 9, el segundo caso si solo hay agujeros en la mitad inferior de la imagen se retorna 6, el tercer caso si se presentan agujeros no conectados en ambas mitades de la imagen se retorna 8, y en caso que el agujero esté al mismo tiempo en ambas mitades de la imagen se pasa al siguiente nivel del árbol de decisión para determinar si es que es un 4 o un 0. Para esto se trazan 2 líneas sobre la imagen (una horizontal y otra vertical), si ambas presentan una cantidad de píxeles de interés mayor a un *threshold* se retorna 4, en otro caso se retorna 0.

En caso de que no se presenten agujeros en la imagen se realiza una operación de contar la cantidad de píxeles de interés en distintas zonas de la *bounding box*, en caso de que la primera fila supere un *threshold* se retorna 7, si la última fila

supera un *threshold* se retorna 2, si no se cumple ninguna de las 2 condiciones se pasa al siguiente nivel del árbol donde la decisión está dada por la diferencia entre la cantidad de píxeles pintados entre la mitad izquierda y derecha de la imagen, si es que dicha diferencia se encuentra sobre un *threshold* se retorna 3, en otro caso se retorna 5.

Para ilustrar de mejor manera el algoritmo se adjunta un diagrama de la solución:

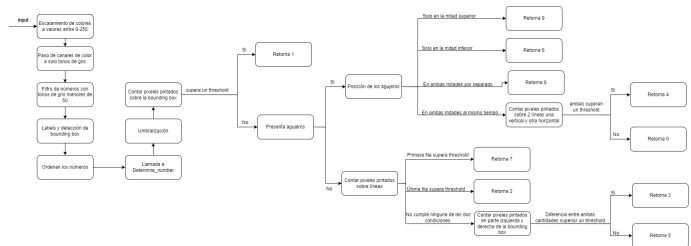


Fig. 1: Diagrama de flujo

## II. EXPERIMENTOS REALIZADOS

Dentro de la experimentación se probaron distintos valores para los *thresholds* usados para filtrar o caracterizar un número de forma generalizada, entre estos se encuentra el valor del filtro inicial, la umbralización, *percentageNumberOne* número usado para determinar si el número es 1, *percentageLine* valor usado para clasificar el 0, 2, 4 y 7, el último valor probado fue *percentageDiff* para determinar si el número era 3 o 5.

### A. Resultados

A continuación se presenta la matriz de confusión obtenida:

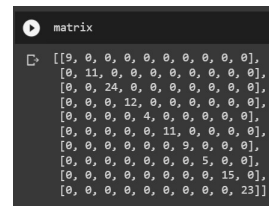


Fig. 2: Diagrama matriz de confusión

Donde el *accuracy* resultante fue del 100%

## III. CONCLUSIONES

La tarea fue una gran instancia para aprender e implementar técnicas para la caracterización de patrones aplicados en la detección de números, cumpliéndose el objetivo propuesto.