

Práctica 2 de Visión Artificial

Detección y reconocimiento de señales de tráfico

Tomás Villalba Ferreiro

1. Pasos generales:

La idea principal se basa en los siguientes pasos:

1. Se deja solo en la imagen los colores azules y rojos.
2. Se aplica algún tipo de filtro para deshacerse del ruido y que la imagen quede limpia.
3. Se aplica un algoritmo de detección de bordes a la imagen.
4. Se busca en la imagen de bordes formas de círculos, cuadrados y triángulos.
5. Se comprueban las características de cada una de las figuras y se clasifican en sus respectivas categorías.

2. Realización:

El primer problema encontrado fue que alguna imagen estaba muy oscura, para solucionar esto se aplicó una ecualización de histograma para aquellas imágenes que tuviesen una mediana de brillo menor a un valor establecido.

Después de este paso procedemos a aplicar una serie de máscaras a la imagen para quedarnos solo con los colores azules y rojos (dados unos rangos de valores establecidos). Un dato importante es que la imagen de colores rojos y la imagen de colores azules se irán procesando por separado, esto facilitará el proceso y dará lugar a menos errores.

El siguiente paso sería el deshacerse del ruido, y mejorar la imagen para su futuro procesamiento, para esto aplicamos la función `medianBlur` para deshacernos de píxeles no deseados que están sueltos. La siguiente idea sería quedarnos no solo con los bordes de color rojo por ejemplo si no con un trozo mas grande de la señal, para esto aplicamos la función de dilatación y una nueva mascara de color blanco, esto ampliará el tamaño de la circunferencia y se quedará solo con la parte que tenga blanco, en un principio el interior de la señal. Se vuelve a aplicar el `medianBlur` para deshacernos de

posible ruido después de la dilatación y aplicación de la máscara y, por último, aplicamos un cierre para que los posibles huecos en las señales se junten.

Cabría destacar que este proceso está encapsulado en una sola función llamada `elimOtherColors`. Hay otra versión llamada `elimOtherColorsSimple` que hace lo mismo, pero solo aplica un filtro de mediana. Estas funciones se aplicarán tres veces a la imagen con distintos parámetros (unos que encuentren mejor señales con huecos, otros que encuentren mejor señales borrosas y otro que encuentre señales claras a simple vista) para conseguir el mejor resultado.

En el siguiente paso se aplica la función de canny para la detección de bordes.

Por último, se buscarán las distintas figuras, círculos en la imagen roja y azul, cuadrados en la azul y triángulos en la roja.

Círculos:

La idea de la función de detección de círculos es aplicar la función `houghCircles` para utilizando parámetros bastante permisivos para luego hacer un cribado por colores y cantidad de círculos encontrados en la misma zona.

En resumen, se hace un primer filtro donde solo nos quedamos con los círculos que tengan al menos otros dos círculos en su cercanía, a no ser que cumplan las condiciones del stop en cuyo caso también nos quedamos con estos círculos (ya que el stop no tiene una forma circular tan perfecta).

Para los círculos que pasen este primer filtro comprobaremos que cumplen con las características de alguna de las señales, Stop, Prohibido, Prohibición, Obligación y se les asigna un nombre según a cuál de las señales pertenezcan.

Por último, nos quedaremos solo con uno de cada clúster de círculos, en concreto con el de mayor tamaño.

Triángulos:

Se utiliza la función `findContours` y nos quedamos solo con aquellos que tienen tres vértices, son aproximadamente equiláteros y tienen un área mayor a un valor.

Los dos tipos de señales triangulares que hay son ceda el paso y de peligro, la diferencia que decidí usar es que las de ceda el paso apuntan hacia abajo (tienen un vértice que tiene una posición en el eje y sustancialmente inferior al resto).

Cuadrados:

Igual que para los triángulos se utiliza la función findContours pero esta vez nos quedamos con los que tienen 4 vértices. Además de esto comprobamos que el área sea mayor a un valor, que los ángulos sean aprox 90 y que todo vértice tenga una pareja en cada eje con una diferencia menor a un valor.

Como ya se mencionó al hablar sobre la función elimOtherColors, este proceso se aplicará tres veces a la imagen, después de cada aplicación se eliminará de la imagen (dejándolo en negro) las señales encontradas para evitar encontrar múltiples veces la misma señal y todos los resultados se irán dibujando en la imagen original.

3.Resultados:

Las celdas vacías representan que no existía tal señal en la imagen.

Imagen	señales	prohibición	ceda	stop	peligro	obligación	indicación	prohibido
0	4/6		1/1			2/3	1/2	
1	3/3	1/1			2/2			
2	2/2		1/1			1/1		
3	2/2			1/1		1/1		
4	4/4					4/4		
5	2/2	2/2						
6	0/2			0/2				
7	2/2			1/1		1/1		
8	2/2		1/1			1/1		
9	2/2					1/0	1/2	
10	5/5					1/1	4/4	
11	3/3					1/1		2/2
Total	31/35	3/3	3/3	2/4	2/2	12/13	6/8	2/2
Precisión	88,57%	100%	100%	50%	100%	92,31%	75%	100%

4.Problemas y posibles mejoras

Los mayores problemas son las señales que están demasiado alejadas, las que están cortadas por la imagen y aquellas que están demasiado desgastadas o con un tono de color raro. No se me ocurre ninguna solución simple usando estos métodos.