

Detección y reconocimiento de señales de tráfico

Nuria Iglesias

January 8, 2024

1 Introducción

Este proyecto se enfoca en la detección y clasificación de diferentes señales de tráfico presentes en imágenes de entornos de carretera. El propósito principal consiste en la identificación de señales de prohibición, peligro, obligación e indicación mediante criterios específicos como forma, color y patrones característicos.

1. Prohibición. Rojas y circulares, “Ceda el paso” o “STOP”.
2. Peligro. Rojas y triangulares.
3. Obligación. Azules y circulares.
4. Indicación. Azules y cuadradas.

El desarrollo se divide en tres etapas, cada una enfocada en aspectos específicos de detección y reconocimiento, consiguiendo de esta manera, aproximarnos a un análisis más detallado de cada señal de tráfico.

2 Esquema global del método

La fase inicial es un **preprocesamiento** de las imágenes de entrada. Se aplica un filtro gaussiano con un núcleo de 3x3 para eliminar el ruido de la imagen y convertimos la imagen al espacio de color HSV, para facilitar la detección de colores rojo y azul. A mayores, se identifican distintas tonalidades de rojo y azul, asegurando así una detección precisa de las señales en cada caso.

La primera etapa del proceso implica **detectar círculos**. Para eso, realizamos la **transformada de Hough** con el método `HOUGH_GRADIENT_ALT`, una variante de `HOUGH_GRADIENT` más precisa. Se realiza sobre la máscara azul o roja, lo que significa que ya hemos realizado el primer filtrado por color y forma. Una vez identificados los elementos circulares, se comprueba su interior, por lo que aquí se divide en dos procesos diferenciados:

1. En el caso de las señales de obligación, los círculos azules, todos son constituidos por flechas de diferentes formas y orientaciones.
 - (a) En una primera instancia, sobre el interior, se realiza una umbralización automática con el algoritmo de Otsu y se seleccionan solamente aquellas regiones que contienen algún píxel. Por consiguiente, se descartan aquellos que no cumplan con un contorno y un área en específico.
 - (b) Se consideró usar plantillas deformables debido a su flexibilidad para adaptarse a las variadas formas de flecha u orientaciones. Sin embargo, debido a la complejidad de crear estas plantillas y al hecho de requerir modelos específicos para cada forma, fue descartada. A mayores, se intentó identificar la forma de triángulo del pico de la flecha, pero esto resultó no dar resultados prácticos viables por lo borrosas que se encontraban algunas señales.
2. En el caso de la detección de señales de prohibición, círculos rojos, se analizó que todas estaban formadas en mayor o menor medida por colores rojos y blancos, además de que todas tenían un círculo interno para delimitar el interior de la señal.

- (a) Comenzamos con la umbralización automática a través del algoritmo de Otsu del interior del círculo detectado, seleccionando regiones que contienen algún píxel.
- (b) Se aumenta el contraste y el brillo del interior de la señal, usando la función `convertScaleAbs`, para que detectar el porcentaje en la imagen de los colores rojos y blancos se realice de manera más sencilla. A la vez, se filtra por sus contornos y su área, para pasar al siguiente paso de filtrado.
- (c) Al detectar que todas tenían un círculo interno, se implementó otra transformada de Hough para el interior de la señal (esta vez con el método `HOUGH_GRADIENT`).
- (d) Como implementaciones fallidas encontramos la intención de añadir una comprobación de los números del interior de las señales, con Pytesseract, una herramienta de reconocimiento óptico de caracteres con la finalidad de "leer" los números en las imágenes. A pesar de haber aplicado técnicas de preprocesamiento, no se logró una lectura precisa, lo que al final limitó su utilidad.

La segunda etapa consiste en **detectar triángulos rojos**. Realizamos directamente la máscara binaria sobre el filtrado de color rojo para detectar los contornos. Más adelante filtramos en base al área y a un contorno en específico. En este caso, se comprobó con el algoritmo de Otsu y sin él, y se tomó la decisión de no usarlo en base a sus resultados. Por último para diferenciar las señales de "Prohibición" ("Ceda el paso") y "Peligro" se realiza un conteo de píxeles rojos de las zonas filtradas, la parte superior de la región de interés se compara con la parte inferior, y se clasifica apropiadamente.

La última etapa es la **detección de cuadrados azules**, se realiza la máscara binaria con el algoritmo de Otsu, a continuación se filtra a través el área y el contorno, y por último se comprueba el interior de la imagen, en una zona más pequeña y centrada, para detectar el color negro existente en las señales, color que ninguna otra señal de color azules contiene en esta práctica.

A mayores, se intentó detectar de manera precisa la forma triangular del interior de las señales pero no fue posible debido a la poca definición de los contornos.

Por último, es relevante mencionar que cada señal detectada se organiza y guarda en una estructura de carpetas, clasificadas entre 'Prohibición', 'Peligro', 'Obligación' e 'Indicación'.



Figure 1: Ejemplo de resultado.

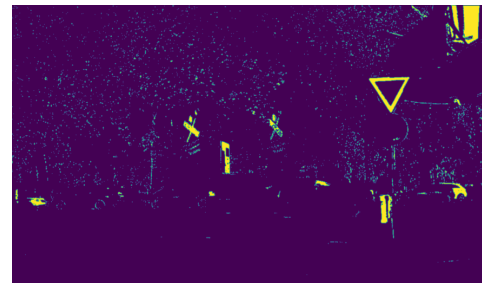


Figure 2: Áreas rojas detectadas.

3 Evaluación

Para evaluar el algoritmo de detección de señales de tráfico y su clasificación se han empleado una serie de métricas:

1. Precisión: Mide la proporción de señales correctamente identificadas (verdaderos positivos) sobre el total de señales identificadas (verdaderos y falsos positivos).
2. Sensibilidad: Evalúa la proporción de verdaderos positivos sobre el total de señales reales (verdaderos positivos y falsos negativos).
3. Similitud: Combina precisión y sensibilidad en una única métrica, proporcionando un balance entre ambas.

$$\text{Precisión} = \frac{\text{Número de señales correctamente detectadas}}{\text{Número de señales correctamente detectadas} + \text{Señales detectadas erróneamente}}$$

$$\text{Sensibilidad} = \frac{\text{Número de señales correctamente detectadas}}{\text{Número de señales correctamente detectadas} + \text{Señales no detectadas}}$$

$$\text{Similitud} = 1 - \left(\frac{\sqrt{(1-p)^2 + (1-s)^2}}{\sqrt{2}} \right)$$

COMPROBACIÓN POR CLASIFICACIÓN

Métricas	Prohibición	Peligro	Obligación	Indicación
Precision	0.93	1	0.81	0.67
Sensibilidad	1	1	1	1
Similitud	0.95	1	0.87	0.77

COMPROBACIÓN POR IMAGEN

Métricas	Imagen 1	Imagen 23	Imagen 40	Imagen 115	Imagen 159	Imagen 171
Precision	0.8	0.6	1	0.67	1	1
Sensibilidad	1	1	1	1	1	1
Similitud	0.86	0.72	1	0.77	1	1

Métricas	Imagen 177	Imagen 214	Imagen 219	Imagen 235	Imagen 262	Imagen 446
Precision	1	1	1	0.67	0.67	1
Sensibilidad	1	1	1	1	1	1
Similitud	1	1	1	0.77	0.77	1

4 Problemas y posibles mejoras

El algoritmo desarrollado puede presentar limitaciones en entornos prácticos distintos debido a su ajuste específico a las imágenes dadas. Además, no ofrece resultados impecables debido a la complejidad de distinguir entre diferentes tipos de señales y .

Para mejorar su funcionamiento, se podría realizar un análisis más exhaustivo de las particularidades de cada señal, considerando aquellas características más específicas. También podría ser beneficioso tener en cuenta la disposición de las señales dentro de la imagen o incluso explorar el uso de redes neuronales convolucionales (CNN) para captar características específicas y potenciar la capacidad de detección de las señales.