

Jornadas de Automática

Algoritmo de reconocimiento de señales de tráfico en entorno simulado.

Feijóo, J.^{a,*}, Rubio, F.R.^a

^aDpto. Ingeniería de Sistemas y Automática, Escuela Técnica Superior de Ingeniería, Universidad de Sevilla, Camino Descubrimientos, 41092, Sevilla, España.

To cite this article: Feijóo, J., Rubio, F.R. 2024. Simulator and algorithm development for traffic sign recognition .
Jornadas de Automática, 45. <https://doi.org/10.17979/ja-cea.2024.45.10806>

Resumen

En este artículo se muestra un sistema de reconocimiento de señales de tráfico en un entorno de simulación desarrollado en ROS y Gazebo. La motivación principal es poder incorporar capacidades de visión a sistemas de navegación más complejos en las fases iniciales de simulación. Se aporta a la comunidad un repositorio 3D con las señales más importantes siguiendo fielmente la normativa española. El algoritmo de percepción sigue una estructura tradicional pero es novedoso el planteamiento como problema de optimización del filtro de color. El clasificador de formas es especialmente rápido usando el primer momento de Hu y una medida de oquedad, y la identificación de serigrafías se lleva a cabo con una red MLP de reducidas dimensiones.

Palabras clave: Aprendizaje automático, Percepción y sensores, Robots móviles, Sistemas/pantallas de información, programación y visión.

Traffic sign recognition algorithm in a simulated environment

Abstract

In this paper, a traffic sign recognition system developed within the ROS and Gazebo frameworks is presented. The main objective is the incorporation of vision capabilities into more complex navigation systems, which are still in the simulation stages. One contribution to the community is a 3D repository of the most relevant traffic signs, thoroughly following the Spanish normative. The structure of the identification algorithm is traditional, nevertheless, the approach to the color discriminator as an optimization problem is groundbreaking. The geometric classifier is especially fast, using the first invariant Hu moment and a cavity descriptor. Finally, sign identification is carried out by a small MLP.

Keywords: Machine Learning, Perception and sensing, Mobile robots, Information displays/system, Programming and Vision.

1. Introducción

Los algoritmos de identificación y reconocimiento de señales viales suponen actualmente una piedra angular en la carrera de la automatización automovilística, y han recibido considerable atención en los últimos años (Wali et al., 2019), (Mathias et al., 2013).

La aparición reciente de simuladores realistas ligados a software robótico han hecho factible emplear escenarios virtuales para las fases iniciales en el desarrollo de sistemas autónomos apoyados en visión (Prabhu et al., 2020), (AbdelHamed et al., 2020) ofreciendo además un entorno controlado para llevar a cabo actividades docentes.

Existen muy diversas estructuras para implementar un sistema de reconocimiento: (Ren et al., 2009) basados en características SIFT o SURF (tasa de acierto del 95 %) , (Kuo and Lin, 2007) empleando RBF, redes neuronales y K-d tree (tasa de acierto de 95.5 %), (Soendoro and Supriana, 2011) formulando el problema para un SVM (tasa de acierto 97 %) entre otros. A pesar de esta gran variedad, un elemento común en los sistemas anteriores es el filtro de color como primera etapa. Este se suele sintonizar heurísticamente, por lo que se considera un área de posible mejora.

En este artículo se comienza introduciendo el entorno de simulación de Gazebo, y se continua con una exposición del

sistema de reconocimiento propuesto, haciendo especial énfasis en el planteamiento del discriminador de color como problema de optimización.

2. Entorno virtual de simulación

El entorno virtual de simulación se describe en archivos *.world* que son interpretados por Gazebo directamente o a través de ROS mediante un *launcher*. El *.world* ubica los objetos y referencia los repositorios donde están sus descripciones, una visualización del mundo creado puede verse en la Figura 1. Las librerías usadas son de la comunidad para la mayoría de los objetos, pero para las señales se ha creado una manualmente, pues no existe en código abierto ninguna que las modele siguiendo las dimensiones de la norma española (derivada de la Convención de Viena y por tanto similares a las del territorio europeo).



Figura 1: Visualización mundo de simulación en Gazebo (989KBytes).

Cada señal se compone en 3 bloques fundamentales: poste, serigrafía anterior y chapa galvanizada, para separar la proyección de texturas (vease la Figura 2). El poste se ha modelado usando un prisma (primitiva de Gazebo) y sus herramientas de proyección. Los otros dos bloques tienen *meshes* idénticos pero difieren en la textura proyectada. Los *meshes* son importados al repositorio en formato *COLLADA* desde Blender, con extensiones *.dae*, como puede observarse en la Figura 3.

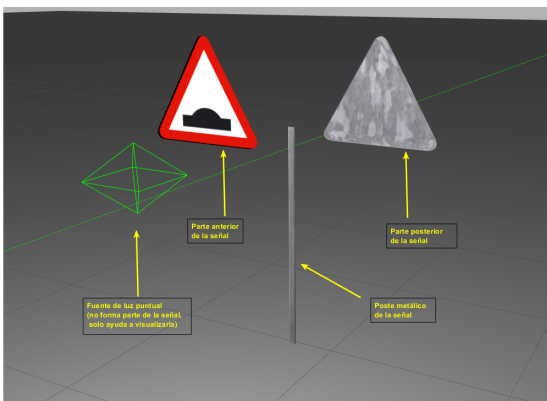


Figura 2: Explosión de una señal en sus 3 bloques internos.

El archivo principal que unifica los *meshes* es el *.sdf*. El *.config* da información complementaria sobre el mantenimiento, el nombre, y la licencia del creador. El archivo *.material* realiza las proyecciones simples en lenguaje de scripts de Gazebo. Las imágenes *.png* de las serigrafías están alojadas en

otro nivel para proporcionar modularidad y facilidad a la hora de realizar futuras modificaciones o ampliaciones.

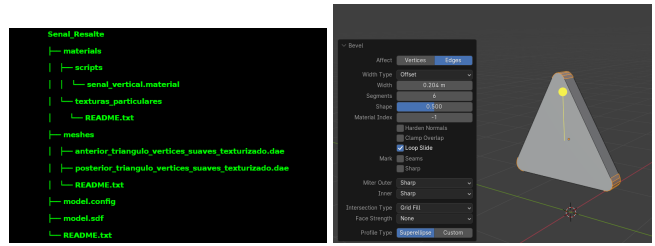


Figura 3: Árbol de directorios de una señal peligro resalte (izquierda). *Mesh* de la misma señal sin las texturas aún proyectadas (derecha).

El repositorio de la señales creadas puede descargarse en <https://github.com/juanfeijoo02/libreria-se-ales-traffic-Gazebo> y se muestra en la Figura 4. Para incluir el repositorio solo es necesario añadir a la variable de entorno `GAZEBO_MODEL_PATH` su ubicación dentro del ordenador: *path_absoluto/biblioteca_senales/senales*.

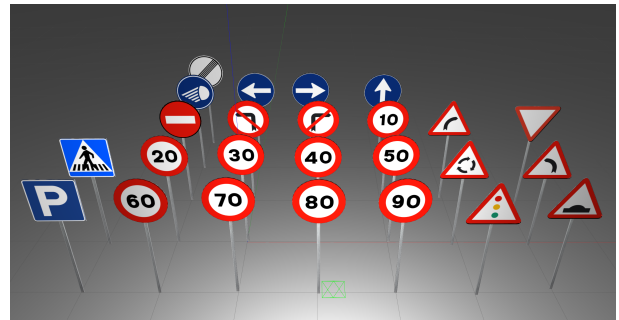


Figura 4: Conjunto de señales modeladas y un foco de luz puntual.

3. Algoritmo de reconocimiento

3.1. Esquema general del programa de percepción

A grandes rasgos el programa de percepción realiza 3 pasos: Filtro de color (p.e resalte del rojo y atenuación del resto de tonos para las señales de prohibición y peligro), clasificación por las formas de los *blobs*, extracción y clasificación de las serigrafías de cada señal. El paso de la degradación de imagen no forma parte del reconocimiento pero sirve para comprobar la efectividad del mismo bajo escenarios no ideales, como puede verse en la Figura 5

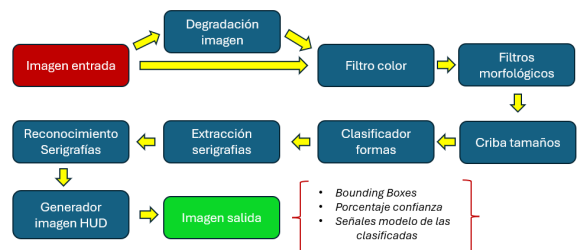


Figura 5: Diagrama general con los pasos del algoritmo de reconocimiento.

3.2. Ajuste óptimo de los filtros de color

Una estructura del optimizador del discriminador de color puede verse en la Figura 6.

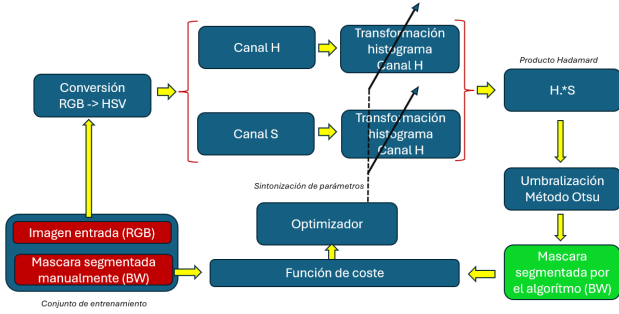


Figura 6: Esquema general del filtro y el optimizador.

El filtro de color requiere primero una conversión de la imagen de entrada RGB al espacio HSV, de este, estaremos interesados en el Hue y en la Saturación, descartando el Valor que es muy sensible a las diferentes condiciones de iluminación. Cada matriz H, S es sometida a una transformación de histograma dada por una función que tendremos que determinar. Se proponen dos, vease la Figura 7. La función definida a trozos (seg-rect), y la pseudo-gaussiana (ps-gau) generada a partir de $f(x) = e^{-x^2}$.

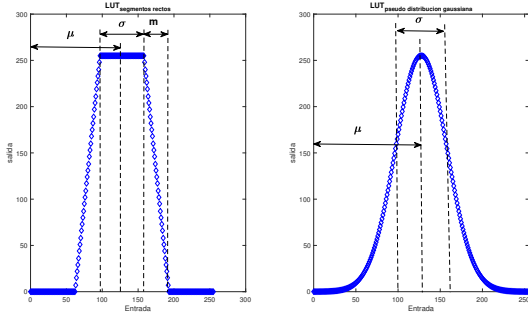


Figura 7: Dos funciones de transformación propuestas en el formato de LUT.

Las matrices H, S transformadas se multiplican elemento a elemento, de esta forma en el array resultante los valores elevados corresponderán a píxeles cuyo Hue y Saturación no han sido atenuados por las funciones de transformación. Esta matriz en escala de grises es binarizada por el método de Otsu para distribuciones bimodales Kiran et al. (2009).

Puesto que se usan dos de estas funciones de transformación, una por canal H, S, se propone un problema de optimización para determinar sus parámetros, que puede resumirse en (1), (2), (3), (4), (5) y (6).

$$\begin{cases} u_{\text{Canal H}} = [\mu, \sigma, m] & u_{\text{Canal S}} = [\mu, \sigma, m] & \text{seg-rect} \\ u_{\text{Canal H}} = [\mu, \sigma] & u_{\text{Canal S}} = [\mu, \sigma] & \text{ps-gau} \end{cases} \quad (1)$$

$$u = [u_{\text{Canal H}}^T; u_{\text{Canal S}}^T] \quad u^{\text{óptimo}} = \arg \max_u \eta(u) \quad (2)$$

Sujeto a restricciones:

$$\begin{cases} \mu \in [0, \text{rango} - 1] & \sigma \geq 1 & m \geq 1 \\ \sigma_{\text{sg}} + 2m \leq \text{rango} - 1 & \text{si seg-rect} \\ \sigma_{\text{pg}} \leq \text{rango} - 1 & \text{si ps-gau} \end{cases} \quad (3)$$

Donde rango es el número de bits dedicado a cada elemento de cada canal (p.e VGA \rightarrow 256). La función de rendimiento propuesta ofrece una interpretación humana en %, es invariante ante los tamaños de imagen y del set de entrenamiento y permite simplificar el problema a la elección del factor de ponderación λ . La matriz p^{auto} es la segmentación automática del algoritmo propuesto, y p^{manual} es una segmentación manual de la imagen original; estos pares constituyen el conjunto de entrenamiento. FIL y COL son las dimensiones de las matrices. α_k las segmentaciones correctas y β_k las incorrectas.

$$\alpha_k = \sum_{i=1}^{\text{FIL}} \sum_{j=1}^{\text{COL}} (p_{ij}^{\text{auto}} \cdot p_{ij}^{\text{manual}}) \quad (4)$$

$$\beta_k = \sum_{i=1}^{\text{FIL}} \sum_{j=1}^{\text{COL}} (p_{ij}^{\text{auto}} \cdot (1 - p_{ij}^{\text{manual}})) \quad (5)$$

$$\eta(u) = \frac{100}{N_{\text{training}}} \sum_{k=1}^{N_{\text{training}}} \frac{\lambda \cdot \alpha_k - \beta_k}{\lambda \cdot \alpha_k} \quad [\%] \quad (6)$$

Con el mismo conjunto de entrenamiento se obtienen mejores resultados de separación cuando las funciones de transformación son las rectas a trozos (más grados de libertad $\text{gdl}=3$), por lo que serán la elección final, en detrimento de las funciones pseudo-gaussianas ($\text{gdl}=2$) que parecen más indicadas para distribuciones Gaussianas, vease la Tabla 1. El hiperparámetro de η toma para este trabajo el valor $\lambda = 5$.

Tabla 1: Valores óptimos en la última iteración del método de bisección.

pseudo-gaussianas				segmentos rectos					
Canal H		Canal S		Canal H			Canal S		
μ	σ	μ	σ	μ	σ	m	μ	σ	m
0	3	230	7	1	4	4	233	18	5
$\eta_{\text{máximo}} \approx 68,298 \%$				$\eta_{\text{máximo}} \approx 69,818 \%$					

Haciendo un análisis de desviaciones respecto al óptimo puede aumentarse la robustez con el mínimo detrimento de eficacia incrementando $\sigma_{\text{Canal H}}$ tal como puede verse en las Figuras 8, 9 y 10. Por comportamiento robusto se entiende que $\mu_{\text{Canal H}}$ puede no coincidir exactamente con el Hue del color a discriminar y aún así segmentarlo correctamente.

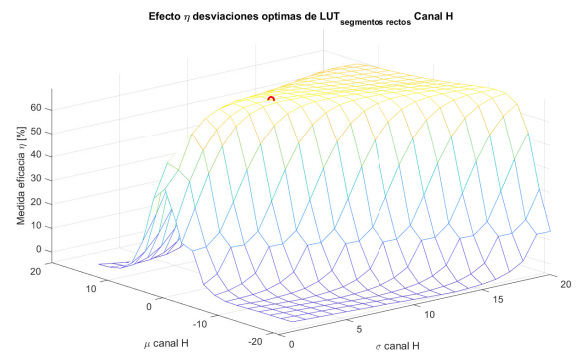


Figura 8: Desviaciones respecto al óptimo de $\sigma_{\text{Canal H}}$ y $\mu_{\text{Canal H}}$.

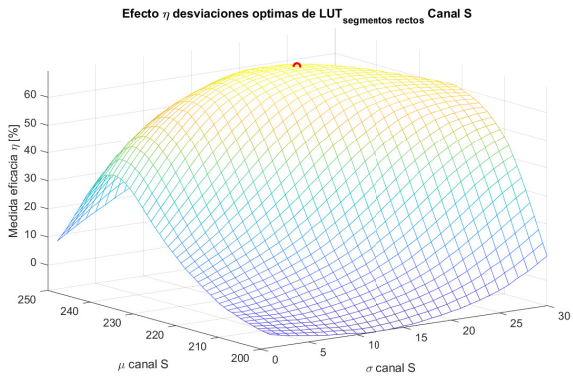


Figura 9: Desviaciones respecto al óptimo de $\sigma_{\text{Canal S}}$ y $\mu_{\text{Canal S}}$. El círculo rojo señala el punto óptimo de máximo rendimiento.

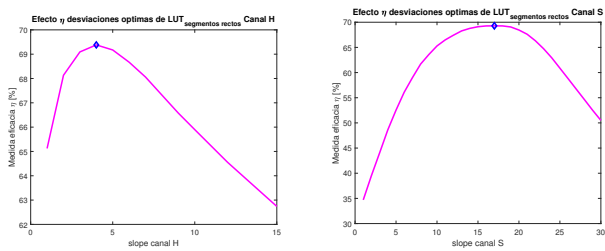


Figura 10: Desviaciones respecto al óptimo de $m_{\text{Canal H}}$ y $m_{\text{Canal S}}$. Los rombos azules señalan el punto óptimo de máximo rendimiento.

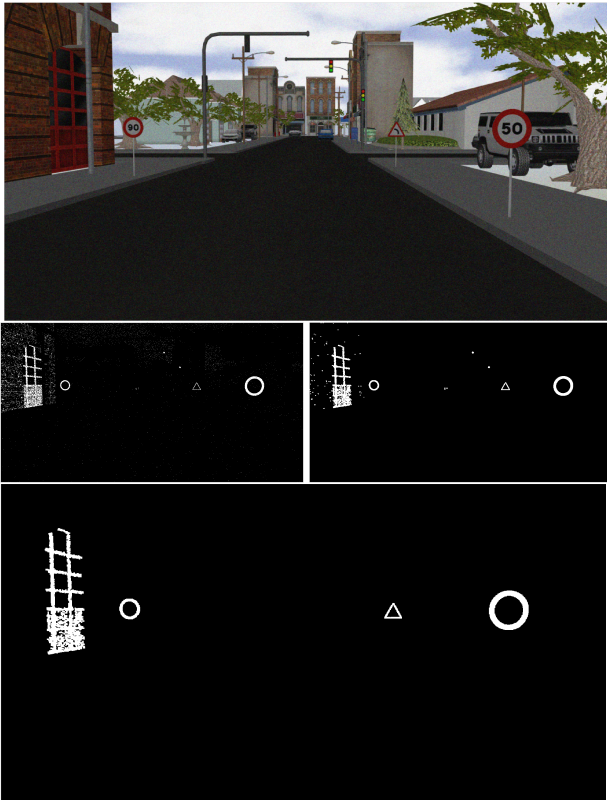


Figura 11: (superior) Imagen original degradada con ruido gaussiano y salt&pepper en cada canal RGB, filtro de media 3x3, y disminución de la resolución. (centro izqda) Imagen binarizada tras el segmentador de color. (centro dcha) Imagen tras las operaciones morfológicas. (inferior) Imagen tras criba de tamaño.

3.3. Clasificador formas

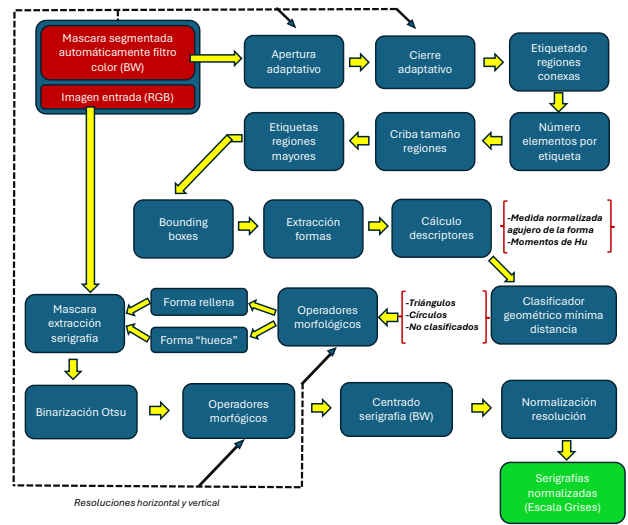


Figura 12: Estructura del clasificador de formas.

El clasificador de formas posee un preprocesamiento (ver Figura 11) para eliminar *blobs* no deseados y mejorar aquellos que pertenecen a las señales para que sean regiones conexas y huecas. El conjunto inicial de descriptores para el clasificador son los 7 primeros Momentos de Hu (Hu (1962)) y una medida de oquedad, pero se hace un análisis de los más relevantes para nuestro problema de clasificación usando una población sintética a partir de muestras reales (ver Figura 13).

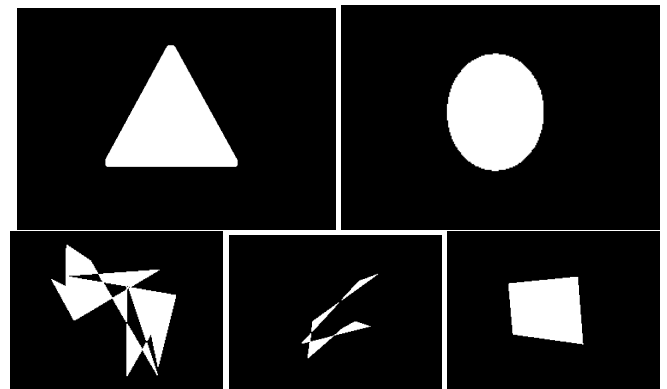


Figura 13: Muestras de la población sintética de las formas rellenas *triangulares* y *circulares* (imágenes superiores), y de la clase *rechazo* (imágenes inferiores). Se juega con rotaciones, escalado, transformaciones homográficas, nº de vértices y posiciones aleatorias sobre muestras reales.

El primer descriptor seleccionado es ϕ_1 , pues presenta una homogeneidad intraclase elevada, suficiente dispersión interclase (a diferencia de ϕ_2, ϕ_3, \dots) y no presenta valores extremos como $\phi_5, \phi_4, \phi_6, \phi_7$ lo cual puede llevar a problemas de robustez y representabilidad, tal como se aprecia en la Tabla 2.

El segundo descriptor cuantifica la oquedad en forma normalizada tal como se muestra en (7).

$$\epsilon = \frac{m_{00}^{\text{forma rellena}} - m_{00}^{\text{forma original}}}{m_{00}^{\text{forma rellena}}} \quad m_{00} = \sum_{i=1}^{\text{FIL}} \sum_{j=1}^{\text{COL}} p_{ij} \quad (7)$$

Ambas clases: *triángulos* y *círculos* deben cumplir la condición $\epsilon \leq 0,25$. El clasificador es de mínima distancia con

Tabla 2: Distribución muestral (media μ y desviación típica σ) de los 7 momentos de Hu para cada conjunto de formas rellenas.

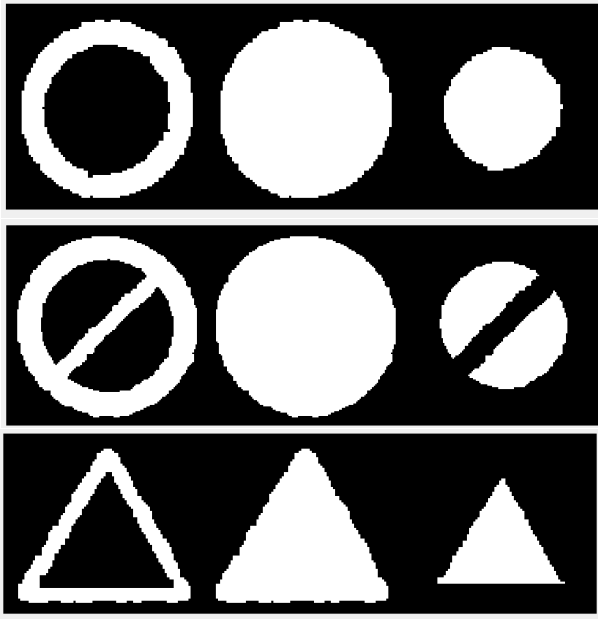
-	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	ϕ_6	ϕ_7
$\mu_{\text{triángulos}}$	$1,932 \cdot 10^{-1}$	$1,466 \cdot 10^{-3}$	$4,326 \cdot 10^{-3}$	$2,813 \cdot 10^{-5}$	$4,368 \cdot 10^{-9}$	$2,268 \cdot 10^{-7}$	$5,020 \cdot 10^{-10}$
$\sigma_{\text{triángulos}}$	$3,041 \cdot 10^{-3}$	$1,452 \cdot 10^{-3}$	$1,041 \cdot 10^{-4}$	$3,579 \cdot 10^{-5}$	$3,133 \cdot 10^{-8}$	$2,582 \cdot 10^{-6}$	$8,473 \cdot 10^{-10}$
$\mu_{\text{círculos}}$	$1,603 \cdot 10^{-1}$	$3,598 \cdot 10^{-4}$	$2,861 \cdot 10^{-8}$	$3,611 \cdot 10^{-9}$	$-9,012 \cdot 10^{-20}$	$-2,265 \cdot 10^{-13}$	$1,311 \cdot 10^{-20}$
$\sigma_{\text{círculos}}$	$1,378 \cdot 10^{-3}$	$5,612 \cdot 10^{-4}$	$3,461 \cdot 10^{-8}$	$5,920 \cdot 10^{-9}$	$2,853 \cdot 10^{-18}$	$7,434 \cdot 10^{-12}$	$2,133 \cdot 10^{-18}$
μ_{otros}	$3,673 \cdot 10^{-1}$	$6,679 \cdot 10^{-2}$	$7,242 \cdot 10^{-1}$	$4,405 \cdot 10^{-1}$	$4,874 \cdot 10^2$	4,114	-4,647
σ_{otros}	$4,015 \cdot 10^{-1}$	$9,496 \cdot 10^{-1}$	$2,282 \cdot 10^1$	$1,403 \cdot 10^1$	$2,122 \cdot 10^4$	$1,494 \cdot 10^2$	$2,080 \cdot 10^2$

prototipos las medias de ϕ_1 para cada clase y aceptando una desviación máxima de 10σ , que de excederse asigna *rechazo*.

Si la clase es *triángulos* o *círculos*, se procede con extracción del símbolo mediante una máscara según (8). Esta se erosiona adaptativamente (en función de la resolución original) para que no haya restos de los bordes de la señal según (9). Ejemplos ilustrativos se adjuntan en la Figura 14.

$$\text{mascara} = (\bar{p}_{\text{original}} \cap p_{\text{rellena}}) \ominus \mathbf{1}_{n \times m} \quad (8)$$

$$n = \text{ceil}\left(\frac{FIL}{200}\right) \quad m = \text{ceil}\left(\frac{COL}{400}\right) \quad (9)$$


 Figura 14: Ejemplos de p_{original} (izqda), p_{rellena} (ctro) y mascara (dcha).

La máscara de extracción se aplica a la imagen RGB convertida en escala de grises según (10).

$$\text{serigrafía}_{ij} = \text{mask}_{ij} \cdot \text{Imagen Gris}_{ik} \quad \forall [i, j] [l, k] \in R \quad (10)$$

Siendo R la región del *bounding box* de la forma n ésima. La imagen extraída con la máscara está en escala de grises por lo que se binariza aplicando el umbral de Otsu según (11).

$$\text{serigrafía BW}_{ij} = \text{serigrafía}_{ij} \leq U_{\text{Otsu}} \quad (11)$$

A continuación las matrices se recortan de tal forma que en las primeras y últimas filas y columnas debe haber al menos un elemento no nulo. Las resultantes se reescalan al tamaño $FIL_{\text{red}} = 30$ y $COL_{\text{red}} = 30$ interpolando con *splines* cúbicos

para minimizar las pérdidas de información (ver 12), con lo que sus elementos vuelven a ser reales $\in [0, 1]$.

$$\text{norm} = \text{rescalado}(\text{serigrafía BW}, FIL_{\text{red}}, COL_{\text{red}}) \quad (12)$$

Ejemplos de la aplicación de las máscaras, binarización, centrado y reescalado se muestran en la Figura 15.



Figura 15: Ejemplos de extracción de las serigrafías de la imagen original, conversión a BW, centrado y normalización de las serigrafías.

4. Identificación serigrafías y resultados

Las matrices con las serigrafías normalizadas se vectorizan y se introducen en una MLP (Multi Layer Perceptron), de una capa oculta, la salida de la última capa posee $n+1$ salidas, siendo $n=17$ el número de señales entrenadas y una salida adicional para el conjunto de descarte. Esta salida predicha se somete a un operador softmax para que el vector de predicción contenga una medida de la confianza de la respuesta, pero puesto que en el entrenamiento se ha condicionado la red para que sus resultados no difieran de $\hat{y} = [0 \dots 0 \ 1 \ 0 \dots 0]^T$ (la posición del 1 indica el tipo de señal detectada) es un paso omisible. Para el entrenamiento de las redes se ha recurrido a un simple algoritmo de descenso de gradiente con momento (Goodfellow et al., 2016). El conjunto etiquetado se ha generado artificialmente a partir de muestras reales (Ishida et al., 2007) (veasé la Figura 16).

Se han probado diferentes funciones de activación: sigmoïdal, RELU y tangente hiperbólica, obteniéndose mejores resultados con la primera, y con los tamaños siguientes en las capas: Capa de entrada: 900×120 , capa intermedia: 120×50 y capa de salida: 50×18 obteniendo una tasa de acierto de 98,667%. La red se ha entrenado durante $N_{\text{epoch}} = 500$ pasos de descenso de gradiente con subconjuntos de $N_{\text{batch}} = 90$ imágenes aleatorias. Redes de mayor tamaño, especialmente en la capa intermedia, ofrecen peores resultados, pues si se prosigue su entrenamiento acontece el fenómeno de *overfitting*, y si se hace una parada temprana su efectividad es inferior.

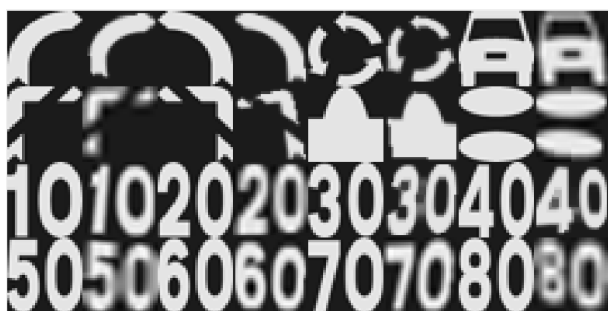


Figura 16: Muestras de la población sintética de entrenamiento junto con sus prototipos. El grupo rechazo no se muestra aquí.

Finalmente el algoritmo traza en las imágenes originales un rectángulo sobre las señales clasificadas: verde si es *circu- lar* y rojo si es *triangular*, muestra un prototipo de la misma y ofrece un índice de confianza en la clasificación, que, aunque no aparece en el interfaz final con el usuario (veasé la Figura 17), permite realizar diagnósticos de funcionamiento internos.

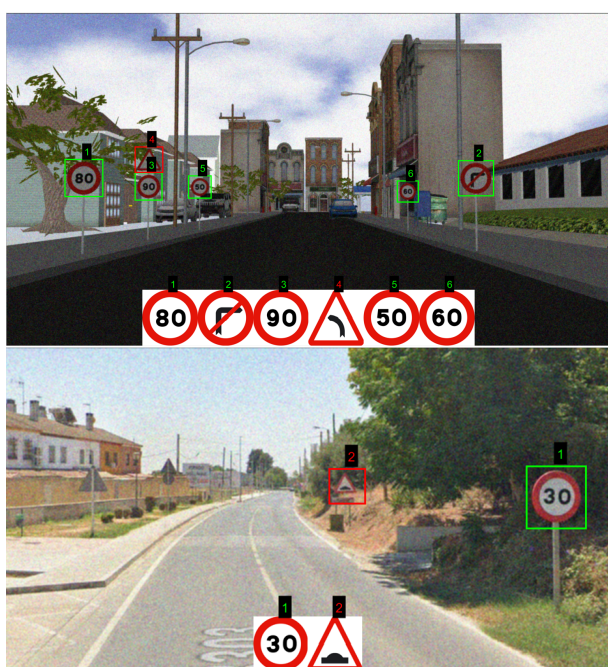


Figura 17: Ejemplos de correcta identificación de todas las señales en imágenes fuertemente degradadas. (arriba) imagen tomada del mundo virtual. (abajo) imagen real de una avenida.

Para evaluar la efectividad del algoritmo, se han llevado a cabo experimentos en distintas partes de la ciudad virtual, en imágenes reales, y a distancias variables. Las tasas de acierto recogidas se muestran en la Tabla 3.

Tabla 3: Estadística de clasificación correcta (CC) en imágenes degradadas.

[%]	Señales de límite de velocidad [Km/h]								
	10	20	30	40	50	60	70	80	90
CC	94	92	86	90	78	72	86	82	92
	Curva peligrosa		Prohibido Girar						
	Dcha	Izqda	Dcha	Izqda					
CC	94	92	86	88					
	Glorieta	Resalte	Semáforos	No adelantar					
	CC	56	78	44	62				

5. Conclusiones

Los resultados son satisfactorios: se ha probado la utilidad y gran capacidad de discriminación del filtro de color propues- to, la eficacia del sistema de reconocimiento bajo condiciones no ideales es aceptable, y el repositorio modelado de señales es fidedigno.

Como trabajo futuro, podría experimentarse con funcio- nes con mayor número de parámetros en el filtro de color, en aras de segmentar varios colores simultáneamente. Con- vendría estudiar para estos casos un algoritmo de optimiza- ción más avanzado (se ha estado usando bisección recursiva), el obstáculo es la no derivabilidad analítica del bloque de bi- narización de Otsu que impide usar descenso de gradiente y otras técnicas diferenciales explícitas. Además se podría ampliar el conjunto de señales reconocibles, mejorar los índices de clasificación correcta e implementar técnicas avanzadas de reconstrucción y reconocimiento para casos con oclusiones o fondos de colores similares.

Agradecimientos

Este trabajo ha sido realizado en el marco de una Beca de Colaboración en el Departamento de Ingeniería de Sistemas y Automática de la Universidad de Sevilla.

Referencias

AbdelHamed, A., Tewolde, G., Kwon, J., 2020. Simulation framework for development and testing of autonomous vehicles. 2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), 1–6. DOI: 10.1109/IEMTRONICS51293.2020.9216334

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press.

Hu, M.-K., 1962. Visual pattern recognition by moment invariants. IRE Transactions on Information Theory 8 (2), 179–187. DOI: 10.1109/TIT.1962.1057692

Ishida, H., Takahashi, T., Ide, I., Mekada, Y., Murase, H., 2007. Generation of training data by degradation models for traffic sign symbol recognition. IEICE - Trans. Inf. Syst. E90-D (8), 1134–1141. DOI: 10.1093/ietisy/e90-d.8.1134

Kiran, C., Prabhu, L. V., V., A. R., Rajeev, K., 2009. Traffic sign detection and pattern recognition using support vector machine. 2009 Seventh International Conference on Advances in Pattern Recognition, 87–90. DOI: 10.1109/ICAPR.2009.58

Kuo, W.-J., Lin, C.-C., 2007. Two-stage road sign detection and recogni- tion. 2007 IEEE International Conference on Multimedia and Expo, 1427–1430. DOI: 10.1109/ICME.2007.4284928

Mathias, M., Timofte, R., Benenson, R., Van Gool, L., 2013. Traffic sign recognition — how far are we from the solution? International Joint Confe- rence on Neural Networks (IJCNN), 1–8. DOI: 10.1109/IJCNN.2013.6707049

Prabhu, N., Min, S., Nam, H., Tewolde, G., Kwon, J., 07 2020. Integrated fra- mework of autonomous vehicle with traffic sign recognition in simulation environment, 514–521. DOI: 10.1109/EIT48999.2020.9208241

Ren, F., Huang, J., Jiang, R., Klette, R., 2009. General traffic sign recogni- tion by feature matching. 2009 24th International Conference Image and Vision Computing New Zealand, 409–414. DOI: 10.1109/IVCNZ.2009.5378370

Soendoro, D., Supriana, I., 2011. Traffic sign recognition with color-based method, shape-arc estimation and svm. Proceedings of the 2011 Interna- tional Conference on Electrical Engineering and Informatics, 1–6. DOI: 10.1109/ICEEI.2011.6021584

Wali, S. B., Abdullah, M. A., Hannan, M. A., Hussain, A., Samad, S. A., Ker, P. J., Mansor, M. B., 2019. Vision-based traffic sign detection and recog- nition systems: Current trends and challenges. Sensors 19. DOI: 10.3390/s19092093