

Aplicación Android para clasificar señalamientos en campus universitario usando aprendizaje de máquina

Elohim Ramírez-Galván, Cesar Benavides-Alvarez,
Carlos Avilés-Cruz, Arturo Zúñiga-López

Universidad Autónoma Metropolitana,
Unidad Azcapotzalco,
México

{a12223800666, cesarbenavides, caviles,
azl}@azc.uam.mx

Resumen. El poder identificar correctamente la señalización en un campus universitario es de suma importancia para estar informado y poderse ubicar y desplazar dentro del mismo. En el presente trabajo se plantea el desarrollo, implementación y pruebas de un sistema clasificador de imágenes en una aplicación para dispositivos móviles Android. El sistema obtiene un vector tridimensional a partir de los canales RGB (Red Green Blue) de la imagen para poder clasificarla mediante la evaluación del vector con clasificadores distintos (Coeficiente de correlación, Bayes, Perceptrón, K-próximos vecinos (KNN)), mismos que devuelven una selección de clase y que el sistema elige la mejor representada. El sistema propuesto tiene una eficiencia de 99.67 % en la correcta clasificación de la señalización.

Palabras clave: Clasificador binario, perceptrón, bayes, kNN, clasificador de imágenes.

Android Application to Classify Signage on University Campuses Using Machine Learning

Abstract. Being able to correctly identify the signage on a university campus is of utmost importance in order to be informed and to be able to locate and move around within it. In this work, the development, implementation, and testing of an image classifier system in an application for Android mobile devices are proposed. The system obtains a three-dimensional vector from the RGB image channels in order to classify it by evaluating the vector with different methods (Correlation Coefficient, Bayes, Perceptron, K-nearest neighbors (KNN)), which return a class selection, and the system chooses the best-represented one. The proposed system has a 99.67% efficiency in the correct classification of the signaling.

Keywords: Binary classifier, perceptron, bayes, kNN, image classifier.

1. Introducción

Dentro del ámbito de la Inteligencia Artificial, y en particular, en el aprendizaje de máquina, el reconocimiento de imágenes es una rama de crecimiento con diversas aplicaciones en múltiples campos. Existen diversas maneras en las que esto se muestra, dependiendo del campo y objetivo. El presente trabajo consiste en el desarrollo, implementación y pruebas de un clasificador de señalizaciones pertenecientes a la Universidad Autónoma Metropolitana Unidad Azcapotzalco (UAM-A), cuyo objetivo es distinguir entre dos tipos de señales (clases) que forman parte de un grafo (como nodos) para la navegación dentro del campus universitario.

De esta manera la navegación móvil proporciona una manera conveniente y eficaz de encontrar direcciones y acceder a una variedad de servicios relacionados con la ubicación, convirtiéndola en una herramienta indispensable en la vida diaria. En este trabajo se presenta un sistema clasificador binario que distingue entre señales de tipo Directorio (Dir) y señales de tipo Punto de Reunión (PR), mostradas en la Figura 1. Cada una de las clases cuenta con características fácilmente distinguibles entre las cuales destaca el color.

El sistema propuesto aprovecha esta característica para diferenciar ambas clases de manera efectiva. El clasificador binario está disponible para dispositivos móviles con sistema operativo Android debido a que, según el Instituto Federal de Telecomunicaciones (IFT) en la Cuarta Encuesta 2020, de Usuarios de Servicios de Telecomunicaciones, más del 80 % de usuarios de telefonía móvil usan este Sistema Operativo, por lo que se pretende que el sistema pueda ser usado por un mayor número de usuarios. El resto del artículo está constituido del estado del arte capítulo 2, la Metodología, descrita en el capítulo 3. En el capítulo 4 se describe la experimentación y resultados; finalmente, las conclusiones y perspectivas son descritas en el capítulo 5.

2. Estado del arte

Existe numerosa bibliografía relacionada al reconocimiento de imágenes, siendo en su mayoría realizada con redes neuronales convolucionales (CNN), que son bien conocidas por los buenos resultados que ofrece, sin embargo, para ello se suele requerir de un amplio conjunto de imágenes, además de numeroso tiempo de entrenamiento y recursos en el equipo de cómputo que utiliza, así como en los amplios conocimientos en el tema para su correcto desarrollo. Es por ello, que la revisión del trabajo relacionado se centra en clasificadores (que no sean CNN) cuyos datos de entrada son imágenes y entre cuyas características principales para la clasificación se encuentre el color.

En [1] se realizó un sistema que permite la detección de enfermedades torácicas a partir de imágenes de Rayos X. Las imágenes pasan por un preprocesamiento para luego discernir entre paciente sano o enfermo. Cuando el resultado es “enfermo”, pasa por 16 clasificadores más, cada uno asociado a una enfermedad en particular, y de la cual se obtiene una probabilidad de que la imagen corresponda a la enfermedad en sí. Al final, el sistema muestra la enfermedad de la probabilidad más alta para la imagen ingresada. Todos los clasificadores están basados en DenseNet. En [4], se presentó un sistema para reconocer señales de tránsito en Tailandia.



Fig. 1. Ejemplo de señalizaciones de PR (izquierdo) y Dir (derecho).

El sistema consta de dos procesos principales, el primero es un clasificador que determina el tipo de señal a la que corresponde de entre 4 posibles, para ello hace uso de las características obtenidas por medio de extracción usando histograma de gradientes orientados (HOG) y descriptor de capas de color (CLD), mediante una máquina de soporte vectorial (SVM) y random forest “bosques aleatorios” (RF) para este propósito. Posteriormente, sigue una etapa de reconocimiento de signo que se encuentra dentro de dicha señal.

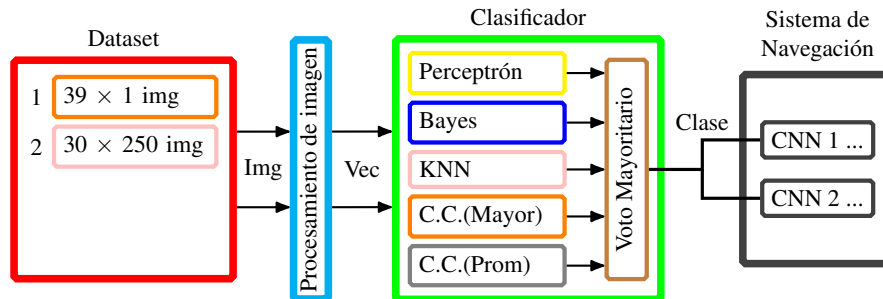


Fig. 2. Modelo de clasificación binario.

En [7], se propuso un método de visión por computadora (usando MATLAB) que permite reconocer frutas, y la idea es que al combinarse con una báscula se pueda obtener el costo respectivo. Esto se realiza a partir de la detección de su forma, color y tamaño. Primero, se realiza un preprocesamiento de la imagen, y una vez extraídas las características de interés, se pasan por diversos clasificadores (KNN, RF, Análisis de datos (DA), SVM), en donde se encontró que KNN tuvo una mejor precisión.

En [5], se presentó un sistema de Visión por computadora, que usando Matlab, distingue entre dos clases de pistaches. Los autores crearon su propio conjunto de datos con una cámara de alta resolución y de las cuales se extrajeron 16 características de interés, mismas que se usaron para la clasificación de los pistaches usando KNN.

En [6], se desarrolló un algoritmo de análisis de imagen para la clasificación de cerezas. Primero se realiza una segmentación de la imagen para dejar únicamente a la fruta, luego se obtiene el histograma de la imagen en formato RGB y Hue Saturation Value (HSV), donde R y H son los componentes que presentan diferencias para la clasificación, y se usan en un clasificador bayesiano para determinar la pertenencia a una de las 5 clases existentes, obteniendo una precisión del 100 %. La implementación se realizó en una computadora con velocidad de procesador de 300 MHz.

En [9], se presentó un clasificador que usa KNN para poder delimitar a la fruta del dragón en la imagen, pues existen diferentes colores para los distintos tipos en diferentes áreas y condiciones de crecimiento, mientras que usa CNN para extraer las características externas de la fruta (como dimensiones y defectos), y finalmente clasificarla entre los 3 grupos posibles.

En [8], se presentó un sistema que detecta las manchas en las hojas de algunas plantas mediante extracción de características como el color y la textura comparando dos clasificadores: SVM optimizado con Bayes y RF. En [3], se propuso un proceso de clasificación que permite determinar el nivel de madurez de la naranja basado en las características de su cáscara (como el color), para ello se utiliza HSV, así como otros métodos para la textura, y siendo clasificados mediante KNN.

En [2], se determinó el nivel de enfermedad de plantas en función de las manchas en sus hojas usando extracción de características como la textura usando Gray-Level Run-Length Matrix (GLRLM) y Gray-Level Occurrence Matrix (GLCM), y color (RGB, HSV, Lab) para desarrollar modelos de clasificación usando SVM, KNN y CNN, y donde se encontró que los mejores resultados fueron para la combinación GLRLM-HSV, clasificados con KNN para obtener los mejores resultados.

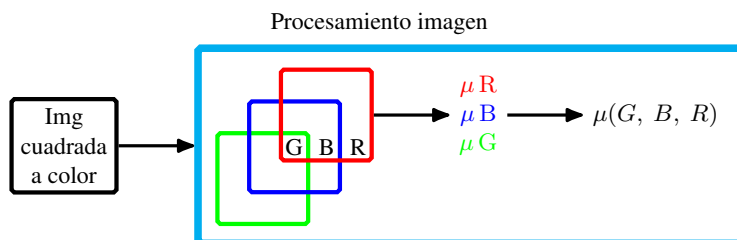


Fig. 3. Procesamiento de imagen de entrada.

3. Metodología

Como ya se mencionó anteriormente, el clasificador binario que se presenta, forma parte de un sistema más grande usado para navegación dentro de la UAM-A. En la Figura 2 se encuentra el esquema del clasificador binario, el cual consiste básicamente en dos etapas, donde la primera es el procesamiento de la imagen, y la segunda corresponde al proceso de clasificación, además del correspondiente conjunto de datos (dataset) necesario para el entrenamiento y evaluación.

3.1. Conjunto de datos

La UAM-A cuenta con un total de 39 señalizaciones de interés (14 PR y 25 Dir) distribuidas en todo el campus, fotografiadas cuidando el enfoque y encuadre adecuado, con un teléfono Xiaomi Poco X3 Pro usando la cámara nativa con tamaño 3000×3000 píxeles, en formato jpg. Este conjunto inicial, fue utilizado para el entrenamiento de los modelos descrito más adelante. Debido a la similitud entre algunas señalizaciones (por ejemplo, algunos edificios cuentan con 2 Dir), estas fueron agrupadas como una misma clase con el fin de simplificar el sistema de navegación, del cual forma parte el clasificador presentado en este trabajo.

De igual forma, algunas otras fueron descartadas debido a la relevancia que tenían, con el mismo propósito. Finalmente quedaron 30 señales diferentes (14 PR y 16 Dir) mostradas en la Figura 1, y que sirven como nodos para el sistema de navegación, y de las cuales se tomaron más de 250 fotografías de cada una, con variaciones de cercanía, iluminación, y perspectiva. Este segundo conjunto de datos, se usó (seleccionando de manera aleatoria entre las imágenes donde la señal no estuviera alejada) para realizar las pruebas una vez teniendo los modelos entrenados.

3.2. Procesamiento de imagen

Las entradas, tanto para el sistema como para el entrenamiento consiste en una imagen cuadrada a color de 1000×1000 píxeles (aunque puede trabajar con diferentes resoluciones), en formato png, la cual es descompuesta en sus 3 canales RGB, y obteniendo el valor promedio de cada canal. Por lo que al final de este proceso se obtiene un vector tridimensional por cada imagen. Este proceso se representa en la Figura 3.

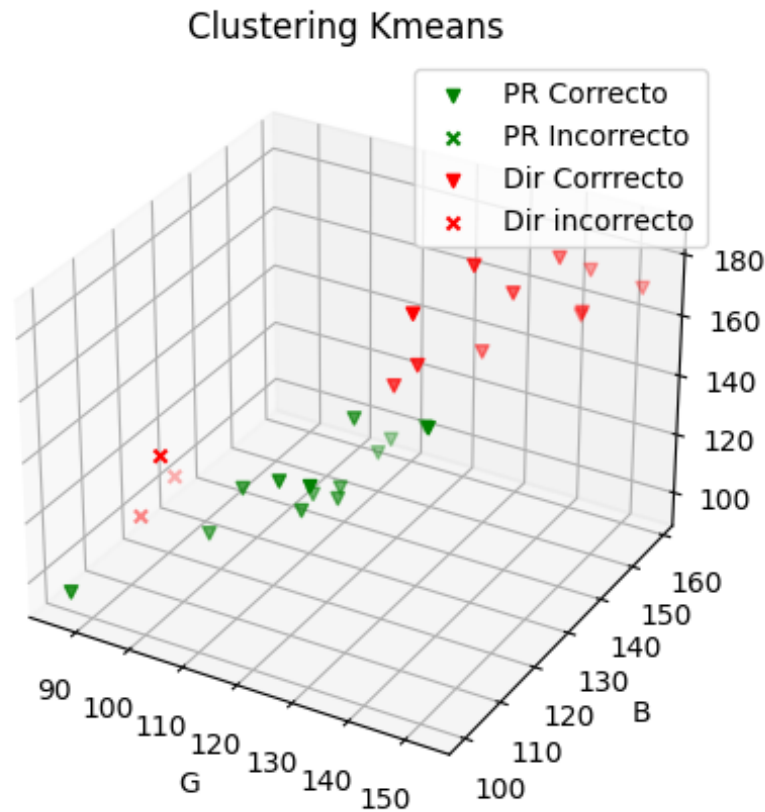


Fig. 4. Agrupamiento usando kmeans.

3.3. Clasificadores

Todos los clasificadores requieren de un entrenamiento a fin de ajustarse a las entradas y resultados deseados. Para realizarlo es necesario un conjunto de datos inicial, que en este caso corresponde a 39 imágenes (una por cada señalización reconocible del modelo de navegación). De estas últimas se descartaron 11 de manera aleatoria para que quedaran ambas clases del mismo tamaño. De modo que al procesar estas imágenes se obtienen 14 vectores para una de las dos clases. Esta misma base sirve para el entrenamiento de los clasificadores, como se indica a continuación:

Perceptrón. Debido a que únicamente son dos clases, se trata de un problema linealmente separable y por tanto, con un único perceptrón. Además, al tratarse de vectores tridimensionales, el perceptrón devolverá la ecuación del plano separatriz entre ambas clases, como se muestra en la Ecuación 1:

$$y = x_1 \times w_1 + x_2 \times w_2 + x_3 \times w_3 + \theta, \quad (1)$$

donde w corresponde a los pesos para cada eje, en este caso, los canales RGB, y x los mismos canales del vector a evaluar.



Fig. 5. Interfaz de la aplicación android.

Y que al evaluar, que el resultado sea mayor o igual que cero, o menor, determinará la clase de pertenencia. El entrenamiento se inició con todos los parámetros en 1 y la base fue normalizada para el entrenamiento, llegando a la ecuación del plano en la iteración 2856. Resultado del entrenamiento: Ecuación del plano separatriz:

$$y = -2,29 B - 4,06 G + 9,07 R - 2. \quad (2)$$

Bayes. La Ecuación 3 para obtener la probabilidad de un vector x en \mathbb{R}^n pertenencia a una clase es la siguiente:

$$P(x/C_k) = (2\pi)^{-n/2} |\Sigma_k|^{-1/2} e^{-1/2(x - \mu_k)\Sigma_k^{-1}(x - \mu_k)^T}. \quad (3)$$

Sin embargo cuando es una clasificación binaria se puede simplificar como se muestra en la ecuación 4:

$$\ln \frac{|\Sigma_1|}{|\Sigma_2|} + (x - \mu_1)\Sigma_1^{-1}(x - \mu_1)^T - (x - \mu_2)\Sigma_2^{-1}(x - \mu_2)^T > 0 \Rightarrow x \in C_1. \quad (4)$$

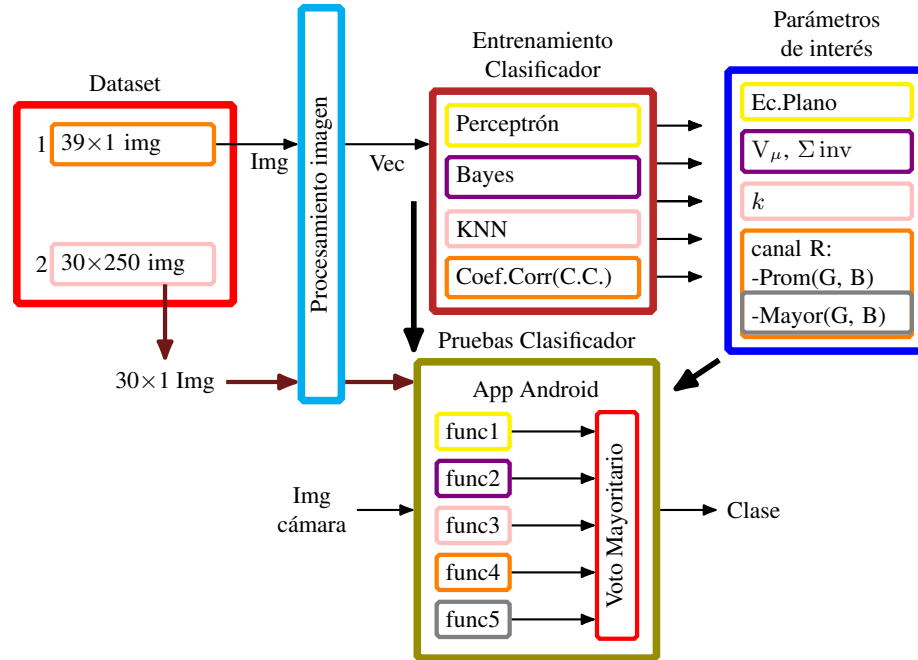


Fig. 6. Interfaz de la aplicación android.

Resultado del entrenamiento: Vectores promedio por clase:

- $\mu_1 = (114,2, 125,0, 120,9)$.
- $\mu_2 = (137,1, 142,8, 168,2)$.

Logaritmo del cociente de determinantes:

$$\ln \frac{|\Sigma_1|}{|\Sigma_2|}. \quad (5)$$

Matriz inversa por clase:

$$\Sigma_1^{-1}, \quad (6)$$

$$\Sigma_2^{-1}. \quad (7)$$

KNN. Para este clasificador se realizó una prueba con $k = 5$, bajo el criterio de k impar \leq a la raíz cuadrada del número de muestras, probándose los KNN de cada subconjunto de atributos posible (G, B, R, GB, BR, GR, GBR), en donde se obtuvieron los mejores resultados (100%) en todos aquellos donde aparece el canal R . Se probó también, calcular la distancia del vector completo (RGB), así como únicamente del canal rojo (R) sin que hubiera diferencias entre ambos. Esto puede apreciarse en las figuras 7 y 5, que aparecen en el clasificador como “KNN” y “-KNNRed” respectivamente. Resultado del entrenamiento: Base de valores promedio:

$$k = 5. \quad (8)$$

Tabla 1. Matriz de confusión por modelo.

		Perceptrón		Bayes		Mayor		Promedio		KNN		Final	
Total	Tipo	PR	Dir	PR	Dir	PR	Dir	PR	Dir	PR	Dir	PR	Dir
14	PR	14	0	14	0	14	0	14	0	14	0	14	0
16	Dir	1	15	1	15	1	15	4	12	1	15	1	15

Coefficiente de Correlación. Se calculó el Coeficiente de correlación entre los atributos y la clase usando la Ecuación 9:

$$R_{ij} = \frac{C_{ij}}{\sqrt{C_{ii} C_{jj}}}, \quad (9)$$

donde se obtuvo que el canal R, es el que está fuertemente relacionado con la clase de pertenencia (como también se puede apreciar en KNN y en la ecuación del perceptrón), por lo que una vez teniendo esta información se revisaron los vectores de la base a fin de encontrar alguna expresión matemática que pudiera realizar la clasificación en función de sus canales, de las cuales resultaron dos: Resultado del entrenamiento: Pertenece a Dir si:

$$R \text{ es } 10\% > \text{prom}(G, B), \quad (10)$$

$$R \text{ es } 5\% > \text{máx}(G, B). \quad (11)$$

K-means. Adicionalmente a los clasificadores anteriores, se realizó clustering usando kmeans para ver si las clases lograban diferenciarse por sí mismas, logrando una precisión mayor al 89%. En la Figura 4 se muestra la gráfica del agrupamiento resultante, con únicamente 3 errores al clasificar Dir, y el 100% para PR.

Voto mayoritario. Finalmente, para tomar la decisión de cuál es la clase a la que pertenece la señal registrada, se ingresa el vector obtenido del procesamiento a todos los clasificadores, que a su vez devuelven una clase. Se toma como clase final de pertenencia aquella que tenga al menos 5 coincidencias.

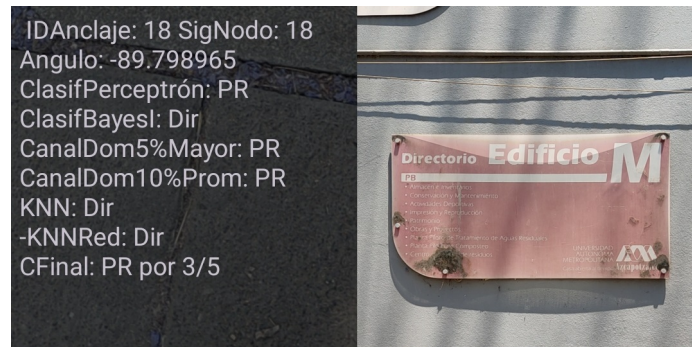
3.4. Aplicación.

El clasificador está implementado en una aplicación Android realizada en Android Studio Giraffe — 2022.3.1 Patch 2. La aplicación cuenta con una pantalla única en la que se visualiza la cámara (previa aceptación de permisos). Esta sección está delimitada con un cuadrado a fin de que la imagen capturada corresponda con las imágenes de entrenamiento, y de este modo el usuario encuadre la señal en este espacio, restringiendo el espacio de píxeles no pertenecientes al señalamiento.

En la parte inferior se ubica un botón con la función de tomar la fotografía, convertir la imagen al vector de colores correspondiente, y pasarlo por las 5 funciones clasificadoras, cuyos resultados individuales y mayoritario se despliegan en un campo de texto, así como la imagen capturada en el campo de imagen correspondiente. En la figura 5 se muestra la interfaz de la aplicación.



(a) Clasificación correcta PR.



(b) Clasificación incorrecta Dir.

Fig. 7. Clasificación de señalización desde aplicación android.

Cada clasificador se realizó a modo de una función independiente que toma como entrada un vector tridimensional y devuelve un valor booleano asociado a la clase de pertenencia. En la Figura 6 se muestra el proceso seguido para la implementación de las funciones (una por clasificador), para todas ellas, se requiere un vector tridimensional como valor de entrada, devolviendo un valor binario correspondiente a la clasificación. Esos valores son los que paran a la sección de Voto mayoritario para su evaluación.

4. Resultados

Una vez obtenidos los resultados del entrenamiento, se realizaron pruebas con imágenes distintas a las del conjunto inicial, obteniéndose una precisión del 96.67%. Para esta se seleccionó aleatoriamente una imagen por clase, de entre las 30 existentes para el sistema de navegación. En la Tabla 1 se detalla la matriz de confusión. Finalmente se implementó el clasificador en una aplicación Android que permite tomar una fotografía y despliega en pantalla el resultado de todos los clasificadores, incluyendo el voto mayoritario. La figura 7 muestra al clasificador operando en la aplicación Android, y se puede apreciar un ejemplo de clasificación correcta, así como una incorrecta.

Con esta prueba, se logró identificar que las señalíticas erróneamente clasificadas corresponden a las del tipo Dir, que se encuentran a la intemperie, y que presentan por tanto desgaste en comparación a las otras. Este desgaste se aprecia principalmente a modo de decoloración que atenúa los tonos rojos de la misma, y que resultan clave para la correcta clasificación. Sin embargo, este problema sólo resulta cuando el encuadre de la señalización no es adecuado.

Si se compara la clasificación errónea de la figura 7, con el ejemplo existente en la figura 5, se puede apreciar que se trata de la misma señalización, pero que en esta última figura, el encuadre se realizó mejor que en la primera, por lo que la clasificación en ese caso es adecuada. Y de este modo, se alcanza la precisión del 100 % en la clasificación. Este hallazgo es importante porque permite establecer el funcionamiento en entornos donde la variación en el desgaste de las señales sea mínima o controlada, e incluso en casos como la UAM-A donde la mayor parte de las señalíticas se encuentran en buen estado, ofrece una precisión en la clasificación bastante aceptable, aún cuando la toma de las imágenes no sea la ideal.

5. Conclusiones y trabajo futuro

Este trabajo presenta el desarrollo, implementación y pruebas de un sistema clasificador binario de imágenes correspondiente a los tipos de señalización (PR y Dir) de la UAM-A, necesario como módulo para un sistema de navegación. Al ser un sistema diseñado para dispositivos móviles, este debe ser ligero, rápido y preciso, por lo que debe ser lo más simple posible. Es por ello que si bien el sistema trabaja con una imagen como entrada, se realiza un procesamiento con la misma hasta obtener un vector tridimensional de ella, mismo que ingresa a diferentes clasificadores ya reducidos a expresiones algebraicas sencillas para obtener un resultado, mismo que será comparado con el de los demás clasificadores a fin de devolver aquel con una mejor representación del mismo.

El sistema de clasificación resultó altamente efectivo, con precisión superior al 99.6 %, únicamente fallando en señalizaciones visiblemente desgastadas, pero en las que, de tomar las imágenes con un encuadre adecuado se alcanza la precisión del 100 %. Para un trabajo futuro se aumentará otro tipo de clasificadores basados en HSV en lugar de RGB pero con el mismo principio presentado, a fin de ver si se encuentran mejoras que puedan disminuir los errores encontrados. También se pretende eliminar el botón para tomar la fotografía, de modo que la predicción se pueda hacer de manera periódica cada cierto intervalo, a fin de poder mejorar las aplicaciones basadas en el clasificador.

Referencias

1. Agarwal, A., L. V., Paduri, A. R., Mabiyan, R., Wattamwar, M. S., L. D., Darapaneni, N.: Detection of thoracic diseases using chest X-ray: A comparative study of binary class and multiclass classification using deep learning. In: Proceedings of the IEEE Pune Section International Conference, pp. 1–6 (2023) doi: 10.1109/punecon58714.2023.10450111
2. Hayit, T., Endes, A., Hayit, F.: KNN-based approach for the classification of fusarium wilt disease in chickpea based on color and texture features. *European Journal of Plant Pathology*, vol. 168, no. 4, pp. 665–681 (2023) doi: 10.1007/s10658-023-02791-z

3. Mentari, M., Rahmad, C., Muchlisin, M. S., Sukmana, S. E.: Classification of siam orange ripeness level using k-nearest neighbors algorithm and features gray level run length matrix. In: IEEE International Conference on Communication, Networks and Satellite, pp. 272–277 (2023) doi: 10.1109/comnetsat59769.2023.10420620
4. Namyang, N., Phimoltares, S.: Thai traffic sign classification and recognition system based on histogram of gradients, color layout descriptor, and normalized correlation coefficient. In: Proceedings of the 5th International Conference on Information Technology, vol. 14, pp. 270–275 (2020) doi: 10.1109/incit50588.2020.9310778
5. Ozkan, I. A., Koklu, M., Saraçoğlu, R.: Classification of pistachio species using improved K-NN classifier. Progress in Nutrition, vol. 23, no. 2, pp. e2021044 (2021) doi: 10.23751/pn.v23i2.9686
6. Reyes, J. F., Contreras, E., Correa, C., Melin, P.: Image analysis of real-time classification of cherry fruit from colour features. Journal of Agricultural Engineering, vol. 52, no. 4 (2021) doi: 10.4081/jae.2021.1160
7. Shakya, S.: Analysis of artificial intelligence based image classification techniques. Journal of Innovative Image Processing, vol. 2, no. 1, pp. 44–54 (2020) doi: 10.36548/jiip.2020.1.005
8. Singh, A. K., Sreenivasu, S., Mahalaxmi, U. K., Sharma, H., Patil, D. D., Asenso, E.: Hybrid feature-based disease detection in plant leaf using convolutional neural network, Bayesian optimized SVM, and random forest classifier. Journal of Food Quality, vol. 2022, pp. 1–16 (2022) doi: 10.1155/2022/2845320
9. Trieu, N. M., Thinh, N. T.: A study of combining KNN and ANN for classifying dragon fruits automatically. Journal of Image and Graphics, vol. 10, no. 1, pp. 28–35 (2022) doi: 10.18178/joig.10.1.28-35