Sistema de Reconocimiento de Señalamientos en Entornos Abiertos para la Orientación de Personas con Discapacidad Visual

Signage Recognition System in Open Environments for the Visually Impaired

Yesenia González[1], Alejandro Millán[2], Yuli Sánchez[3], Claudia Ortiz[4], Miguel Alemán[5], Carlos Hernández[6]

Recibido: Julio 2020 Aceptado: Diciembre 2020

Resumen.– En este trabajo se presenta el desarrollo de un prototipo que permite identificar señalamientos específicos a través de técnicas de visión artificial. Utiliza una etapa de segmentación de imagen basada en el algoritmo de superpíxeles SLIC, seguida de una etapa de reconocimiento y clasificación de señalamientos basada en redes neuronales convolucionales que se ha implementado en un ordenador de placa reducida (SBC). El prototipo informa al usuario de la identificación de estos señalamientos a través de un mensaje de audio enviado a un auricular, cuenta con un módulo GPS que obtiene la ubicación donde se reconoció el señalamiento y se almacena para ofrecer al usuario notificaciones sobre señalamientos cercanos. Las pruebas fueron realizadas con señalamientos a escala 1:2 en entornos abiertos, con luz de día. El prototipo pretende ser un apoyo para que personas con discapacidad visual puedan desplazarse en entornos abiertos urbanos. Tiempos de procesamiento y desempeño del prototipo son reportados. Si bien la implementación en el ordenador de placa reducida seleccionado hace inviable su uso por los tiempos de operación, se demuestra la funcionalidad del sistema.

Palabras clave: Señalamientos, algoritmo SLIC, redes neuronales convolucionales, ordenador de placa reducida, GPS, personas con discapacidad visual.

Summary.– *This work presents the development of a prototype that allows identifying specific signs through artificial vision techniques. It uses an image segmentation stage based on the SLIC superpixel algorithm, followed by a sign recognition and classification stage based on convolutional neural networks and has been implemented in a simple-board computer (SBC). The prototype informs the user of the identification of these signs through an audio message sent to headphones, it has a GPS module that obtains the location where the sign was recognized and is stored to offer the user notifications about nearby signs. The tests were performed with 1: 2 scale signs in open spaces, with natural light. The prototype is intended as a support for visually impaired people to move in open urban environments. Processing times and prototype performance are reported. Although the implementation in the selected simple-board computer makes its use unfeasible due to operating times, the functionality of the system is demonstrated.*

Keywords: *Signs, SLIC algorithm, convolutional neural networks, simple-board computer, GPS, people with visual disabilities.*

[1] Doctora en Ciencias, UPIITA - Instituto Politécnico Nacional, ygonzalezn@ipn.mx ORCID iD: 0000-0003-2370-4660

[2] Ingeniero, UPIITA - Instituto Politécnico Nacional, amillanj1500@alumno.ipn.mx, ORCID iD: 0000-0002-0789-1112

[3] Ingeniera, UPIITA - Instituto Politécnico Nacional, yuliyulivani@gmail.com, ORCID iD: 0000-0002-9744-4614

[4] Ingeniera, UPIITA - Instituto Politécnico Nacional, alejandra.ortiz1404@gmail.com, ORCID iD: 0000-0002-5314-9488

[5] Doctor en Ciencias, CNMN - Instituto Politécnico Nacional, maleman@ipn.mx, ORCID iD: 0000-0001-6234-8406

[6] Maestro en Ciencias, UPIITA - Instituto Politécnico Nacional. hernandeznc@ipn.mx, ORCID iD: 0000-0002-2417-9020

1. Introducción.– La discapacidad, según la Organización Mundial de la Salud (OMS) es un término general que abarca deficiencias en alguna función corporal de un individuo y que conlleva a limitaciones para que este pueda realizar acciones, por lo que se limita su participación en la sociedad donde vive [1]. En México, en un comunicado de prensa realizado por el Instituto Nacional de Estadística y Geografía (INEGI, por su acrónimo), 7.7 millones de personas de 5 años o más presentan discapacidad [2]. El INEGI ha desarrollado una estructura de clasificación de tipo de discapacidad, siendo uno de los tipos, las discapacidades sensoriales y de la comunicación y a su vez, un subgrupo de estas, son las discapacidades para ver [3]. Las discapacidades para ver se refieren a la pérdida total de la visión, a la debilidad visual (personas que solo ven sombras o bultos) y a limitaciones que no pueden ser superadas con el uso de lentes. Según el INEGI, de la población en México de 5 años o más con discapacidad, 39.6 % presenta discapacidad para ver [2].

Cuando existe una limitación visual, desplazarse de un lugar a otro o conocer nuevos entornos presenta un desafío, lo que puede comprometer su seguridad. La movilidad o capacidad para desplazarse con independencia implica el aprendizaje de técnicas que permiten a la persona con discapacidad visual caminar en línea recta, seguir referencias, cruzar calles y utilizar el transporte público [4]. De acuerdo con una publicación especial de la Unión Latinoamericana de Ciegos (ULAC) en el 2018, los temas presentados seguían abordando los problemas de las personas con discapacidad para el acceso a la tecnología [5]. Circunstancias de carácter económico, cultural y político: limitaciones económicas para la compra de productos de apoyo, no contar con políticas públicas o que ni siquiera se considere como algo fundamental para las personas con discapacidad, que cuenten con las herramientas y destrezas para ser autónomos e independientes.

Tradicionalmente se han desarrollado sistemas de ayuda electrónica para la movilidad de personas con discapacidad visual. Se han propuesto bastones en los que se integran sistemas de detección de obstáculos por ultrasonido o láser e informan al usuario mediante tonos musicales o incluso mediante señales tacto-vibrátiles [6-8]. Otro campo explorado ha sido utilizar técnicas de visión artificial para reconocer los señalamientos de interés [9-11]. Con los avances en años recientes de técnicas de visión artificial basadas en redes neuronales de aprendizaje profundo [12-13], se han propuesto trabajos que utilizan estas arquitecturas, algunos de ellos operando en ambientes en interiores, ambientes controlados y/o fondos claros [14] y otros, para ambientes en exteriores, como lo que se busca en este trabajo [15-17].

Para el desarrollo del prototipo propuesto, se realizó una encuesta a un grupo de estudiantes con ceguera pertenecientes al Comité Internacional Pro Ciegos IAP ubicado en la colonia Santa María la Ribera en la Ciudad de México [18]. Dicha encuesta se enfocó a la factibilidad de uso de un sistema de reconocimiento de señalamientos para su desplazamiento en ambientes urbanos nuevos, el tipo de accesorio donde preferirían que estuviera montado, si preferían que estuviera como aplicación en un teléfono celular inteligente o que fuera un dispositivo específico, el costo que podrían pagar por un equipo de esta naturaleza. Datos relevantes como la dificultad para manejar teléfonos celulares inteligentes y el costo que podrían pagar restringen entonces la propuesta de solución.

Este trabajo propone el desarrollo e implementación de un sistema portátil de bajo costo capaz de reconocer una selección de señalamientos en entornos abiertos y con luz de día, con el fin de facilitar la orientación y movilidad de las personas con discapacidad visual cuando se desplazan por nuevos entornos. El sistema no requiere de un entrenamiento por parte del usuario para su uso. Realiza notificaciones de audio mediante audífonos 3.5 mm, de esta forma el usuario no depende de audífonos especialmente diseñados para el sistema, que limitarían su uso, sino que puede emplear audífonos comerciales. Otra característica importante que presenta el proyecto es la de

ofrecer notificaciones sobre señalamientos cercanos que hayan sido identificados anteriormente, por lo que se utiliza un módulo de Sistema de Posicionamiento Global (GPS) [19].

2. Trabajos relacionados.– Con el objetivo de apoyar en la autonomía de las personas con discapacidad visual, se han desarrollado diversos trabajos que utilizan técnicas de visión artificial. Con respecto a las técnicas de visión artificial basadas en redes neuronales de aprendizaje profundo, un tipo de redes actualmente muy utilizada para el procesamiento de imágenes son las redes neuronales convolucionales (CNN), que como su nombre indica, realiza una operación de convolución entre la entrada de la red (comúnmente un arreglo multidimensional de datos) y diferentes filtros o kernels (arreglos multidimensionales de parámetros). Existen varias arquitecturas estándar, tales como AlexNet [20] o VGGNet [21], dichas arquitecturas proponen diferentes tipos de parámetros y capas. Mientras que para AlexNet el tipo de filtros usados suele ser grande (11×11), la red VGGNet propuesta en [21] utiliza un tamaño de filtros menor (3×3). En [15], al igual que el trabajo aquí propuesto, tienen limitación en el hardware a utilizar, por lo que proponen y prueban diferentes arquitecturas basadas en la red VGGNet, pero disminuyendo el número de capas y filtros a utilizar; también disminuyen el tamaño de las imágenes de entrada ($32 \times 32 \times 3$).

Los trabajos desarrollados en [15] y [16], proponen una etapa previa a la red neuronal convolucional, donde se lleve a cabo una búsqueda de regiones de interés en la imagen de entrada y serán estas regiones las que ingresen posteriormente a la red neuronal convolucional para el reconocimiento y clasificación del señalamiento. En [15], se utiliza el algoritmo de segmentación "Crowcut" propuesto en [22], mientras que en [16] utilizan una combinación de el algoritmo de Histograma de Gradientes (HOG) [23] y el algoritmo "Speed Up Robust Features" (SURF) [24]. En contraste, en [17], la red neuronal convolucional utilizada recibe las imágenes de entrada, pero en su etapa de entrenamiento, a las imágenes utilizadas se les aplica un proceso de etiquetado en las regiones de interés.

3. Metodología.– El sistema debe de identificar 27 señalamientos (ver Figura I) cuya selección fue realizada con base en dos criterios: Las necesidades de un grupo estudiantes con ceguera entrevistados y en aquellos señalamientos que es posible encontrar disponibles actualmente en la Ciudad de México. El sistema es apto solo para residentes de la Ciudad de México debido a que las normativas que rigen los señalamientos varían dependiendo de la localidad.

El término señalamiento o señalamiento vertical se refiere a aquellas señales construidas en tableros con leyendas y pictogramas fijadas en postes, marcos y otras estructuras. Según su propósito, estas señales se clasifican en: señales restrictivas, señales preventivas, señales informativas, señales turísticas y de servicios. Tienen como función reglamentar, informar y advertir acerca de las condiciones de rutas, direcciones y destinos donde transitan los usuarios, esto con la finalidad de salvaguardar la seguridad de estos. Cada señalamiento debe cumplir con características predefinidas en lo referente a forma, diseño, color, dimensión, tamaño y forma de letras, pictograma y símbolo, estas características se encuentran reglamentadas en la Ciudad de México por la Secretaría de Comunicaciones y Transportes a través del Manual de Señalización Vial y Dispositivos de Seguridad [25].

Figura I.- Señalamientos a reconocer.

Sobre la propuesta de solución, esta también fue influenciada por el grupo de personas encuestadas, como ya se mencionó en la Introducción. La solución propuesta es un sistema portable capaz de reconocer señalamientos en entornos abiertos, a través de técnicas de visión artificial y haciendo uso de un ordenador de placa reducida (Raspberry Pi 3). Las imágenes son obtenidas desde una cámara conectada al ordenador de placa reducida (SBC), el cual tiene a su vez conectado un Arduino UNO con un módulo GPS, que permite conocer la ubicación del usuario al momento de reconocer un señalamiento, con la finalidad de ofrecer al usuario notificaciones auditivas a través de un audífono sobre señalamientos cercanos a su ubicación. En la Figura II se puede observar la arquitectura propuesta para el sistema.

La descripción de la función que realiza cada uno de los componentes físicos del sistema se describe a continuación:

- Cámara: Dispositivo a través del cual se realiza la captura del video.
- Raspberry Pi 3: Dispositivo donde se lleva a cabo la detección y clasificación del señalamiento, así como la comparación de las ubicaciones almacenadas y obtenidas del GPS. En este dispositivo se encuentra alojada la base de datos utilizada para el almacenamiento de los señalamientos identificados con anterioridad, así como los mensajes pregrabados que se reproducen en los audífonos.
- Módulo GPS: Módulo a través del cual se obtiene la ubicación actual del usuario.
- Arduino UNO: Dispositivo que recibe la ubicación del usuario (latitud y longitud), del módulo GPS y la envía al ordenador de placa reducida para su comparación y almacenamiento.
- Audífonos: Dispositivos a través de los cuales se reproduce el mensaje de audio seleccionado.

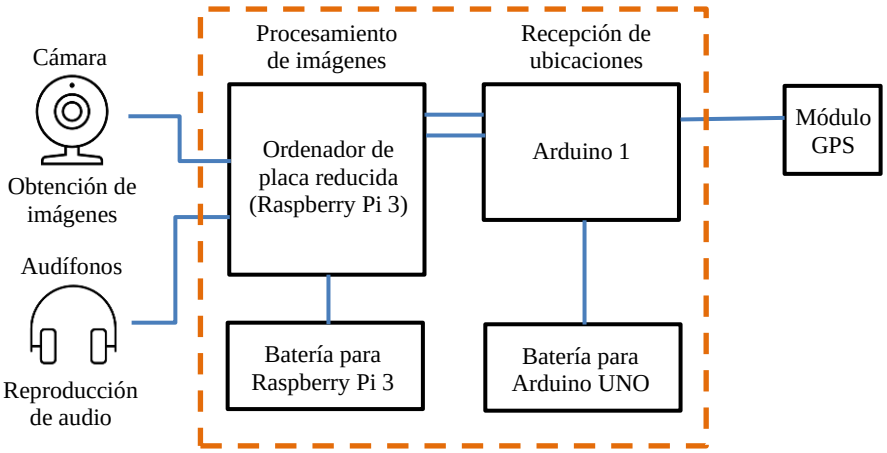


Figura II.- Arquitectura del sistema de detección de señalamientos.

El funcionamiento del sistema se describe a continuación: el usuario inicia el sistema, posteriormente la cámara va realizando la captura de imágenes, el ordenador de placa reducida (SBC) las recibe y realiza la detección y clasificación de un señalamiento siempre y cuando este se encuentre en la lista de señalamientos definidos para el sistema. Una vez identificado el señalamiento, se selecciona un mensaje de audio que contiene el tipo de señalamiento y este mensaje se reproduce en los audífonos para que el usuario pueda escucharlo. También se obtiene la ubicación donde se encuentra el usuario que envía el módulo GPS a través del Arduino Uno y se compara con las ubicaciones donde se identificó algún señalamiento con anterioridad (almacenadas en la base de datos). Si se encuentra alguna coincidencia de ubicación entre el usuario y los datos almacenados, se selecciona el mensaje de audio que contenga el tipo de señalamiento vinculado a la ubicación almacenada, en caso contrario, se actualiza la base de datos con la ubicación del usuario, así como el tipo de señalamiento identificado.

A continuación, se detallan cada una de las etapas del sistema:

3.1. Ubicación.– La ubicación del usuario es obtenida mediante el módulo de GPS Ublox NEO-6M, que se comunica a través del puerto serial UART. Esta presentación modular es compatible con algunas plataformas como Arduino, Raspberry Pi o Laptop. Algunos de los parámetros que mide son latitud, longitud, velocidad y altitud. El módulo GPS se conecta a una placa de Arduino UNO.

Para obtener latitud y longitud del usuario, se hace uso de la biblioteca TinyGPS [26] que permite almacenar los valores de latitud y longitud en variables de tipo flotante. Estas variables se convierten a variables de tipo caracter y se concatenan para enviar a la tarjeta Raspberry una única cadena de texto. El envío de la cadena de texto a la Raspberry se realiza de manera alámbrica a través del bus I^2C (*Inter-Integrated Circuits*), donde la Raspberry es el maestro y el Arduino el esclavo. La Raspberry recibe la cadena que a su vez se divide en dos cadenas de longitud determinada. Posteriormente estas cadenas se asignan a latitud y longitud respectivamente, y se realiza una conversión de variables de tipo cadena (*string*) a flotante para realizar las comparaciones de distancias en búsqueda de señalamientos cercanos.

3.1.1. Búsqueda de señalamientos cercanos.– Para la búsqueda de señalamientos cercanos, primero se obtiene la longitud y latitud del usuario a través del Arduino y se asigna a una variable "ubicación 1". Posteriormente se obtienen cada una de las longitudes y latitudes de los señalamientos almacenados en la base de datos y se van asignando a "ubicación 2". Para cada una de las ubicaciones obtenidas de la base de datos se realiza el cálculo de la distancia entre la "ubicación 1" y la "ubicación 2". La distancia entre las ubicaciones dadas se obtiene en metros y se asigna a una variable distancia. Cada una de las distancias obtenidas se almacena en un arreglo que es ordenado de menor a mayor, y se selecciona la primera distancia en el arreglo que se encuentre entre 10 y 30 metros. De acuerdo con el señalamiento identificado, se reproduce el audio correspondiente.

3.1.2. Almacenamiento de la ubicación.– Una vez que el algoritmo de procesamiento identifica un señalamiento, se realiza una inserción en una tabla de la base de datos, de la ubicación actual del usuario (latitud y longitud) y el tipo de señalamiento identificado.

3.2. Obtención y procesamiento de la imagen.– A través de la cámara se capturan las imágenes para después ser analizadas en busca de señalamientos, este dispositivo debe ser colocado sobre el usuario en conjunto con todo el sistema, por lo que se busca un tamaño reducido y bajo peso, sin que esto se vea reflejado en una baja resolución, además debe ser compatible con sistemas operativos basados en Linux. Por sus características, se seleccionó el modelo *Camera Module v2*.

En esta etapa se lleva a cabo el proceso de adquisición y adecuación de las imágenes para el reconocimiento y clasificación de los señalamientos como se muestra de manera general en la Figura III. Debido a la consideración de utilizar dispositivos portátiles de bajo costo, para el sistema de visión se buscó aprovechar el desempeño que ofrecen las redes neuronales convolucionales, pero buscando arquitecturas de red con pocas capas, para lograr su ejecución en los dispositivos seleccionados. Uno de los principales trabajos consultados es el de Zanetti [15], que propone una red convolucional de 12 capas para el reconocimiento de señales de tráfico. A diferencia de otras redes convolucionales como la mencionada en [17], que a partir de una imagen de entrada de forma automática realizan la identificación de la región donde se encuentran los objetos de interés, la red utilizada en [15] recibe como imagen de entrada la región de interés, por lo que es necesaria una etapa preliminar de búsqueda de la región de interés, que en la Figura III aparece como la etapa de segmentación.

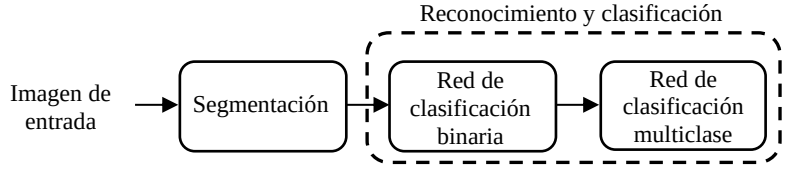


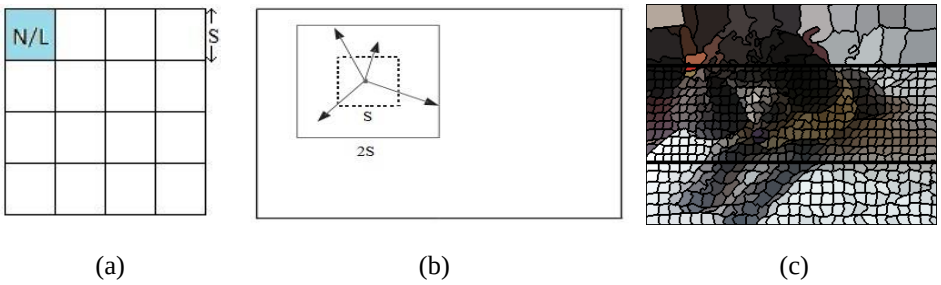
Figura III.- Proceso general de la etapa de obtención y procesamiento de imágenes.

3.2.1. Segmentación.– Una vez adquirida la imagen en formato de color RGB, con una resolución VGA de 640×480 píxeles y con el objetivo de encontrar regiones de interés dentro de la imagen donde pueda estar localizado algún señalamiento, se propuso utilizar un algoritmo de superpíxeles [27-28]. El término de superpíxel fue introducido en 2003 [29] y se refiere al agrupamiento de píxeles con rasgos similares dentro de una imagen. Los superpíxeles proporcionan una manera conveniente de realizar una caracterización focal de parámetros en una imagen digital. Uno de los algoritmos investigados fue SLIC (*Simple Linear Iterative Clustering*) [30-31]. Este es un algoritmo de segmentación que genera superpíxeles a partir del color y la proximidad de píxeles

semejantes. El número aproximado de superpíxeles de tamaño semejante que se desean obtener se determina a través del parámetro de entrada L. Dicho algoritmo segmenta la imagen de entrada en L mallas cuadradas de tamaño fijo, para una imagen de N píxeles, el tamaño aproximado de cada región de la malla sería de N/L píxeles; cada región está espaciada $S = \sqrt{N/L}$ píxeles. Los parámetros N, L y S pueden observarse en la Figura IV.a; posteriormente se localiza el centro de cada una de estas regiones y se calculan los gradientes de todos los píxeles vecinos a este para mover el centro al vecino con el menor gradiente, esto evita que el centro de alguno de los superpíxeles se localice en un borde.

Posteriormente, cada uno de los píxeles se asocia con el centro más cercano en una región de $2S \times 2S$ alrededor del centro localizado como se muestra en la Figura IV.b. Para realizar esta agrupación se utiliza la distancia entre píxeles y la diferencia entre los colores que representan.

Una vez que los píxeles están asociados al centro más cercano, estos se ajustan para obtener el vector medio de todos los píxeles asociados y se calcula un error residual. Esto se repite hasta que el error converge. Finalmente, con los píxeles que permanecen aislados se forma una conectividad a través de un algoritmo de componentes conectadas. El resultado final de la segmentación mediante el algoritmo SLIC se puede observar en la Figura IV.c., que presenta 3 valores distintos de L.



(a) (b) (c)

Figura III.- (a) Representación de una segmentación con $L = 16$ mallas de una imagen. (b) Agrupación de un superpixel. (c) Aplicación del algoritmo SLIC para la segmentación de una imagen con 3 valores distintos de L.

3.2.2. Reconocimiento y clasificación de señalamientos.– Para esta etapa del sistema se emplearon redes neuronales artificiales de aprendizaje profundo [12-13]. El aprendizaje profundo se considera un subcampo de los modelos de aprendizaje de máquina, el cuál utiliza distintas estructuras de redes neuronales para lograr el aprendizaje mediante capas sucesivas de representaciones cada vez más significativas. Un tipo de redes de aprendizaje profundo se conoce como redes convolucionales (CNN, por sus siglas en inglés) y se caracterizan por ser un tipo especializado de red para procesar datos en arreglos estructurados, ya sea en una dimensión (formando un vector), o en el caso de imágenes, en dos dimensiones (arreglo de píxeles).

De acuerdo con la Figura III, previo a la etapa de reconocimiento y clasificación de señalamientos está la etapa de segmentación. Se propuso una red neuronal convolucional de clasificación binaria que analiza cada segmento de imagen generado por la etapa de segmentación y lo clasifica como señalamiento o no. Cada segmento se adecuó a una resolución de 64×64 píxeles. La arquitectura propuesta de esta red está basada en un ejemplo del libro de F. Chollet [13] y en [15]. La Tabla I muestra la arquitectura de la red de clasificación binaria propuesta.

Capa	Entrada	Salida	Función de Activación
Convolucional 1	64, 64, 3	64, 64, 32	ReLU
Reducción 1	64, 64, 32	32, 32, 32	
Convolucional 2	32, 32, 32	32, 32, 64	ReLU
Reducción 2	32, 32, 64	16, 16, 64	
Convolucional 3	16, 16, 64	16, 16, 128	ReLU
Reducción 3	16, 16, 128	8, 8, 128	
Aplanado	8, 8, 128	8192	
Desactivación aleatoria	8192	8192	
Densidad	8192	512	ReLU
Densidad	512	1	Sigmoide

Tabla I.- Parámetros de la red de clasificación binaria.

Una vez que la red de clasificación binaria detecta si el segmento analizado es un señalamiento, este segmento ingresa a una red de clasificación multiclase. La capa de salida se compone de 27 neuronas, ya que se desean reconocer 27 tipos de señalamientos. La Tabla II muestra la arquitectura de la red neuronal de clasificación multiclase. A diferencia de la red de clasificación binaria que utiliza la función Sigmoide en la capa de salida, para la red de clasificación multiclase se eligió la función de activación SoftMax.

Capa	Entrada	Salida	Función de Activación
Convolucional 1	64, 64, 3	64, 64, 32	ReLU
Reducción 1	64, 64, 32	32, 32, 32	
Convolucional 2	32, 32, 32	32, 32, 64	ReLU
Reducción 2	32, 32, 64	16, 16, 64	
Normalización 1	16, 16, 64	16, 16, 64	
Convolucional 3	16, 16, 64	16, 16, 128	ReLU
Reducción 3	16, 16, 128	8, 8, 128	
Normalización 2	8, 8, 128	8, 8, 128	
Aplanado	8, 8, 128	8192	
Desactivación aleatoria	8192	8192	
Densidad	8192	256	ReLU
Densidad	256	128	ReLU
Desactivación aleatoria	128	128	
Densidad	128	27	SoftMax

Tabla II.- Parámetros de la red de clasificación multiclase.

En el caso de que una imagen adquirida por el sistema no contenga señalamientos o bien la red de clasificación binaria no detecte señalamiento en todos los segmentos analizados, no se ejecuta la red de clasificación multiclase.

3.3. Selección y reproducción del mensaje de audio.– Esta etapa consiste en seleccionar mediante una sentencia el audio adecuado al tipo de señalamiento, ya sea detectado o cercano. En este trabajo, los mensajes de audio que se reproducen para notificar al usuario se generaron mediante un sistema de conversión de texto a voz (CVT).

Un sistema de conversión texto a voz (CTV) es un sistema que convierte una entrada de texto en una salida en forma de señal de audio cuyo contenido se corresponde con el mensaje del texto de entrada [32]. Es decir, son sistemas que permiten la conversión de textos en voz sintética. Los conversores de texto-voz son conocidos con las siglas CTV o por las siglas en inglés TTS (Text To Speech).

Se utilizó el sistema CVT del sitio web https://soundoftext.com, con archivos en formato mp3, haciendo uso de voz en idioma español (México). Estos audios se almacenan en la Raspberry, donde se lleva a cabo su reproducción. La salida del audio es a través de los audífonos alámbricos 3.5 mm.

4. Implementación y pruebas del sistema.– Los algoritmos se ejecutaron en una computadora personal de tipo Asus TP501U con un procesador Intel Core i7-6500U CPU @ 2.50GHz, GPU NVIDIA 940MX, 12 GB de memoria RAM. El lenguaje de programación utilizado para el desarrollo de este trabajo fue Python.

4.1. Creación del banco de datos.– Una tarea previa al reconocimiento y clasificación de las imágenes de entrada al sistema es la creación de un banco de datos de imágenes que servirá para el entrenamiento, prueba y validación de los modelos matemáticos. Para este fin, se fabricaron señalamientos a escala 1:2 de los señalamientos originales, que cumplen con las normas del Manual de Señalización Vial y Dispositivos de Seguridad de la Secretaría de Comunicaciones y Transportes, en cuanto a colores y posición para entornos urbanos abiertos, lo que permitió realizar las pruebas dentro de las instalaciones universitarias. Además de que se observaron diferentes problemáticas para la obtención de imágenes de señalamientos en las calles de la Ciudad de México, ya que muchos señalamientos no cumplen con las reglas de señalización y/o se encuentran en mal estado.

Para las redes neuronales convolucionales utilizadas en el sistema para la detección y clasificación de los señalamientos, se realizaron dos bancos de datos, uno para la red de clasificación binaria con un total de 12,616 imágenes y otro para la red de clasificación multiclase con un total de 6,281 imágenes.

Cada uno de los bancos de datos se realizó tomando fotografías de los 27 señalamientos propuestos para las imágenes positivas de la de clasificación binaria y las 27 categorías de la red de clasificación multiclase, así como del escenario de pruebas para las imágenes negativas de la red de clasificación binaria.

Dichas fotografías fueron tomadas en entornos abiertos, con luz de día tanto con teléfonos celulares como con la cámara seleccionada para el sistema. Después, todas las fotografías se exportaron al software Matlab, donde se realizó el etiquetado de pequeñas secciones o regiones de interés (ROIs, por sus siglas en inglés) que contienen señalamientos para las categorías positivas u objetos que se pueden encontrar en el escenario de pruebas para la categoría negativa. El etiquetado de las ROIs se realizó a través de una herramienta GUI de Matlab llamada Training Image Labeler.

Posteriormente, mediante un script de Matlab, todas las ROIs se recortaron y se almacenaron en formato PNG con una resolución de 64 x 64 píxeles. En la Figura V se pueden observar algunas de las imágenes almacenadas con el nombre de la categoría a la que pertenecen.

Para la red de clasificación binaria se utilizaron 6308 imágenes para la categoría "Positivas" y 6308 imágenes para la categoría "Negativas". La Tabla III muestra el número de imágenes de entrenamiento para cada una de las 27 clases de la red de clasificación multiclase.

Figura V.- Ejemplos de ROIs etiquetadas como imágenes "positivas" para la red de clasificación binaria.

Categoría	Total de imágenes	Categoría	Total de imágenes	Categoría	Total de imágenes
Bicicleta	*165*	Información	145	Tienda	152
Escolar	409	Metro	151	Trolebús	299
Peatones	404	Monumento	175	Alarma	357
Autobús	402	Mujeres	234	Alto	518
Baños	245	Zona Segura	197	Hospital	150
Cafetería	158	Basura	*281*	Prohibido	238
Discapacitados	155	Restaurante	150	Emergencia	220
Entrada	175	Taxi	150	Evacuación	236
Hombres	154	Teléfono	150	Reunión	211

Tabla III.- Banco de datos para la red de clasificación multiclase.

4.2. Etapa de segmentación.– Como se mencionó en la sección 3.2.1, SLIC es un algoritmo de segmentación que genera superpíxeles a partir del color y de la proximidad de píxeles semejantes. La Figura VI muestra una de las imágenes del banco de datos segmentada con el algoritmo SLIC. Puede apreciarse que el señalamiento dentro de la imagen queda contenido dentro de uno de los superpíxeles. Con base a pruebas realizadas, el número de segmentos propuesto para el sistema fue $L = 30$. El tiempo promedio de procesamiento de la etapa de segmentación es de 409.5 ms.

Figura VI.- Imagen segmentada con el algoritmo SLIC, con L=30.

4.3. Etapa de reconocimiento y clasificación de señalamientos.– Para el entrenamiento de las redes neuronales convolucionales se utilizó la biblioteca de código abierto TensorFlow. Otra biblioteca utilizada fue Keras, que utiliza a TensorFlow como back-end. En la sección 4.1 se detalla la cantidad de imágenes utilizadas. El entrenamiento de ambas redes se realizó desde cero. Para la red de clasificación binaria se asignó 60 % de los datos para entrenamiento, 10 % de los datos para prueba y 30 % para validación. Para la red de clasificación multiclase, se asignó 60 % de los datos para entrenamiento, 20 % de los datos para prueba y 20 % para validación.

La Tabla IV muestra los parámetros de entrenamiento utilizados para la red de clasificación binaria y para la red de clasificación multiclase [33]. La Figura VII.a muestra la arquitectura de la red de clasificación binaria y la Figura VII.b muestra la arquitectura de la red de clasificación multiclase, ambas desplegadas desde Python.

TIPOS DE PARÁMETROS	RED DE CLASIFICACIÓN BINARIA	RED DE CLASIFICACIÓN MULTICLASE
INICIALIZACIÓN DE PESOS:	Glorot initialization method	Glorot initialization method
PÉRDIDA DE ENTRENAMIENTO:	Binary Crossentropy	Categorical Crossentropy
ACTUALIZACIONES:	RMSprop	SGD and Nesterov momentum
MOMENTO:	-	0.9
TAMAÑO DE LOTE:	32	128
DESACTIVACIÓN ALEATORIA:	-	20 %

Tabla IV.- Parámetros de entrenamiento de la red de clasificación binaria y la red de clasificación multiclase.

En la Figura VIII se muestran las curvas de desempeño de la red de clasificación binaria, usando las métricas de “exactitud” y “pérdida” (ver Tabla IV), para los datos de entrenamiento y validación. De igual manera, en la Figura IX se muestra las curvas de desempeño de la red de clasificación multiclase, para los datos de entrenamiento y validación [34]. Para ambas redes se utilizaron 200 épocas de entrenamiento.

Con base a las curvas obtenidas para ambas redes de las métricas de exactitud y pérdida, para la red de clasificación binaria se observa una ligera oscilación en las curvas del set de validación.

Pruebas más exhaustivas serían necesarias para mejorar el desempeño de la red (introducir capas de normalización, desactivación aleatoria, etc.). Con respecto a la red de clasificación multiclase, se observa un desempeño adecuado, ya que tanto las curvas de entrenamiento y de validación tienden de manera estable hacia el 100 % (en el caso de la exactitud), siendo ligeramente más baja la curva referente al set de validación. Un análisis similar puede hacerse de las curvas de pérdida. Para la red de clasificación multiclase, se incluye una matriz de confusión [34], [35], que nos brinda un análisis más detallado del comportamiento de cada clase del sistema. Se ha observado una relación en la cantidad de falsos negativos o positivos con las clases de menor número de muestras.

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 64, 64, 32)	896
max_pooling2d_1 (MaxPooling2	(None, 32, 32, 32)	0
conv2d_2 (Conv2D)	(None, 32, 32, 64)	18496
max_pooling2d_2 (MaxPooling2	(None, 16, 16, 64)	0
conv2d_3 (Conv2D)	(None, 16, 16, 128)	73856
max_pooling2d_3 (MaxPooling2	(None, 8, 8, 128)	0
flatten_1 (Flatten)	(None, 8192)	0
dense_1 (Dense)	(None, 512)	4194816
dense_2 (Dense)	(None, 1)	513

Total params: 4,288,577
Trainable params: 4,288,577
Non-trainable params: 0

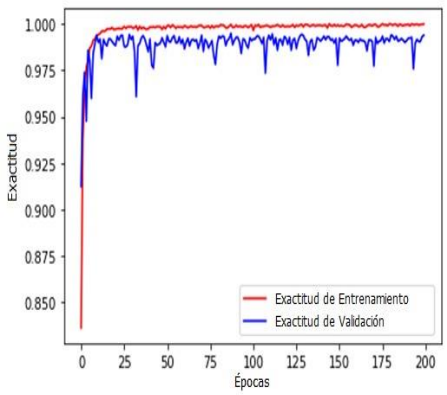
(a)

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 64, 64, 32)	896
max_pooling2d_1 (MaxPooling2	(None, 32, 32, 32)	0
conv2d_2 (Conv2D)	(None, 32, 32, 64)	18496
max_pooling2d_2 (MaxPooling2	(None, 16, 16, 64)	0
batch_normalization_1 (Batch	(None, 16, 16, 64)	256
conv2d_3 (Conv2D)	(None, 16, 16, 128)	73856
max_pooling2d_3 (MaxPooling2	(None, 8, 8, 128)	0
batch_normalization_2 (Batch	(None, 8, 8, 128)	512
flatten_1 (Flatten)	(None, 8192)	0
dropout_1 (Dropout)	(None, 8192)	0
dense_1 (Dense)	(None, 256)	2097408
dense_2 (Dense)	(None, 128)	32896
dropout_2 (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 27)	3483

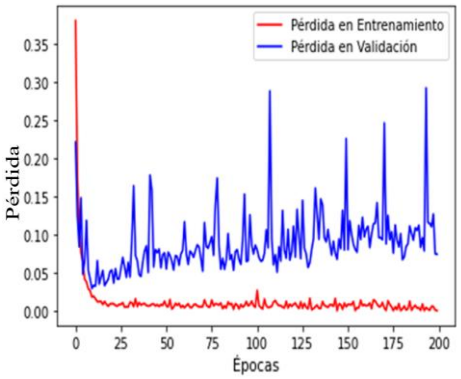
Total params: 2,227,803
Trainable params: 2,227,419
Non-trainable params: 384

(b)

Figura VII.- (a) Despliegue de la arquitectura de la red de clasificación binaria y (b) Despliegue de la arquitectura de la red de clasificación multiclase.

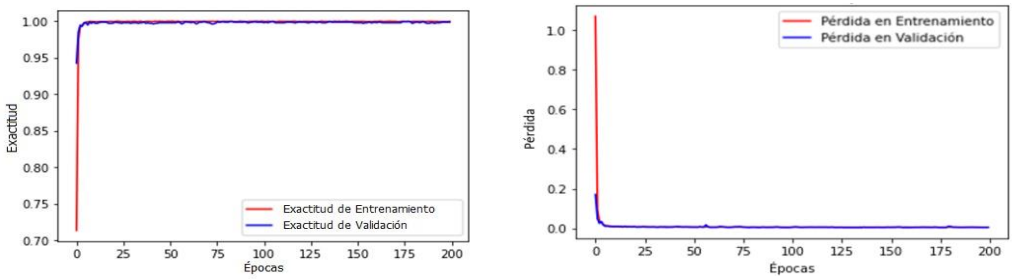


(a)



(b)

Figura VIII.- (a) Gráfica de "exactitud" y (b) gráfica de "pérdida" para los datos de entrenamiento y validación de la red de clasificación binaria.



(a) (b)

Figura IX.- (a) Gráfica de "exactitud" y (b) gráfica de "pérdida" para los datos de entrenamiento y validación de la red de clasificación multiclase.

Valor predicho \ Valor real	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
1	71																										
2		104																									
3			78				1									5							2				
4				46																							
5					37																						
6						30																					
7					8		26																			3	
8								31																			
9									36																		
10										33		1															
11						3					78																
12										2		46															
13													31					1									1
14														27								3					
15															29									3			
16					10											25											
17																	29								1		
18																		46								2	
19									3		3								80	1							
20									5											50							
21																					30						
22														3								39					
23																							28	10			
24																								17			
25																									29		
26			1				2																			55	
27																											36

(a)

1	Alarma	10	Entrada	19	Peatones
2	Alto	11	Escolar	20	Prohibido
3	Autobus	12	Evacuación	21	Restaurante
4	Baños	13	Hombres	22	Reunión
5	Basura	14	Hospital	23	Taxi
6	Bicicleta	15	Información	24	Telefono
7	Cafeteria	16	Metro	25	Tienda
8	Discapacitados	17	Momunmento	26	Trolebus
9	Emergencia	18	Mujeres	27	Zona Segura

(b)

Figura X.- (a) Matriz de confusión de la red de clasificación multiclase. (b) Especificación de señalamiento para cada número de clase (de la 1 a la 27).

Como parte de la investigación, la base de datos utilizada para el entrenamiento y validación de las redes de clasificación binaria y de clasificación multiclase propuestas en este trabajo, también fueron implementadas en otras arquitecturas de redes neuronales convolucionales del estado del arte. Para la selección de dichas arquitecturas, una de las restricciones importantes fue la resolución mínima permitida de las imágenes de entrada, que, en nuestro caso, se tienen imágenes de 64×64 píxeles. La Tabla V y la Tabla VI muestran parámetros de entrenamiento y desempeño de las redes NASNet [36], DenseNet [37] y ResNet [38], entrenadas para detectar la presencia o no de un señalamiento (clasificación binaria) y para la identificación del señalamiento (clasificación multiclase), respectivamente.

Arquitectura	**Resolución de imágenes de entrada**	**Total de parámetros**	**Exactitud Entrenamiento**	**Pérdida Entrenamiento**	**Exactitud Validación**	**Pérdida Validación**
NASNet Mobile	64 × 64	4,271,830	49.17 %	69.37 %	59.78 %	69.31 %
DenseNet201	64 × 64	18,325,826	49.69 %	69.40 %	58.26 %	69.50 %
ResNet50	64 × 64	23,591,810	50.63 %	69.68 %	40.62 %	73.07 %
Red de clasificación binaria propuesta	64 × 64	4,288,577	99.99 %	7.39 %	99.84 %	0.44 %

Tabla V.- Parámetros y desempeño de arquitecturas del estado del arte para la detección o no de señalamiento.

Arquitectura	**Resolución de imágenes de entrada**	**Total de parámetros**	**Exactitud Entrenamiento**	**Pérdida Entrenamiento**	**Exactitud Validación**	**Pérdida Validación**
NASNet Mobile	64 × 64	4,298,255	99.91 %	0.18 %	99.68 %	0.8 %
DenseNet201	64 × 64	18,373,851	99.91 %	0.16 %	99.92 %	0.34 %
ResNet50	64 × 64	23,643,035	99.97 %	0.07 %	99.84 %	0.64 %
Red de clasificación multiclase propuesta	64 × 64	2,227,803	99.95 %	0.46 %	99.84 %	0.44 %

Tabla VI.- Parámetros y desempeño de arquitecturas del estado del arte para la identificación del tipo de señalamiento.

Con respecto a los resultados obtenidos de las implementaciones en las arquitecturas de redes neuronales convolucionales del estado del arte para realizar una clasificación binaria, se observó un bajo desempeño en las tres arquitecturas seleccionadas, siendo contrastante con el desempeño de la red de clasificación binaria aquí presentada.

Sin embargo, las mismas arquitecturas adaptadas para una clasificación multiclase, 2 de ellas presentaron una exactitud menor a la reportada por la arquitectura aquí propuesta y una de ellas presentó mayor exactitud. Todas ellas con una exactitud arriba del 99 %, tanto para la etapa de

entrenamiento como de validación. Cabe resaltar que el número total de parámetros de la arquitectura propuesta es menor a los reportados en las arquitecturas del estado del arte, lo que podría significar una optimización en el uso de recursos para aplicaciones en sistemas embebidos.

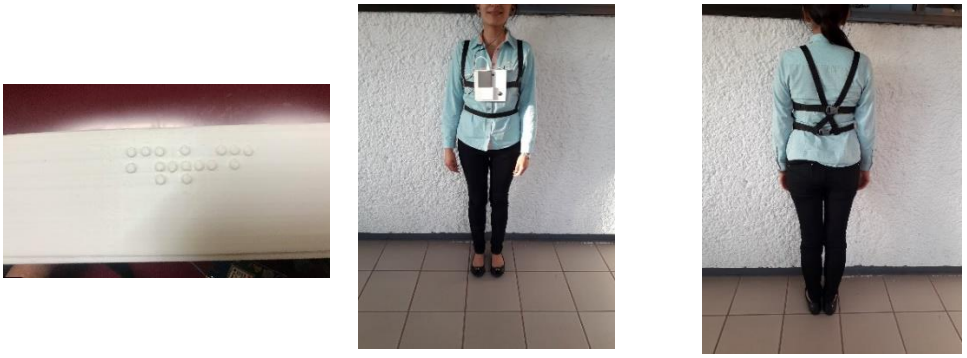
4.4. Implementación del sistema en el ordenador de placa reducida.– El sistema se implementó en una tarjeta Raspberry Pi 3, que cuenta con una memoria externa microSD de 32 GB, en la cual se instaló el sistema operativo UBUNTU MATE 16.04.5. Dicha instalación se realizó a través del software Win32 DiskImager. La Tabla VII presenta las bibliotecas instaladas y sus versiones.

Librería	**Versión**	**Biblioteca**	**Versión**
Python 3	3.5.2	Numpy	1.16.2
Keras	2.2.2	Matplotlib	3.0.3
Skimage	0.14.0	Scipy	0.17.0
Open CV	3.2.0	Geopy	1.20.0
Tensorflow	1.8.0	SQLite	3.11.0

Tabla VII.- Librerías instaladas en la tarjeta Raspberry Pi 3.

4.5. Pruebas al sistema.– El sistema logró ser implementado en la tarjeta Raspberri Pi 3, sin embargo, el uso de recursos limita la velocidad de procesamiento, obteniéndose tiempos de entre 3-5 minutos para cada imagen, lo que a su vez limitó utilizar una cantidad de imágenes robusta para las pruebas al sistema, por lo que los resultados obtenidos no se presentan, al no ser representativos.

Con respecto al diseño de la carcasa del prototipo, se diseñó un sistema de sujeción ajustable a diferentes usuarios, quedando el prototipo a la altura del pecho de la persona. La Figura XI.a muestra la parte superior de la carcasa diseñada que incluye la palabra “arriba” en Braille, las Figuras XI.b y XI.c muestran el montaje del sistema sobre el usuario.



(a) (b) (c)

Figura XI.- Sistema de detección de señalamientos. (a) Vista superior de la carcasa, (b) Montaje frontal del sistema y (c) montaje posterior del sistema.

6. Conclusiones.– Tomando como base la optimización de recursos de hardware, se desarrolló e implementó un prototipo para la detección de señalamientos en entornos urbanos abiertos, para la asistencia a personas con discapacidad visual. Esta propuesta tenía el objetivo de desarrollar un prototipo de bajo costo, pero a su vez, que pudiera incluir técnicas recientes de visión artificial tales como las redes neuronales convolucionales, para lograr robustez en ambientes no controlados. Se propuso un sistema en cascada donde la imagen de entrada pasa por un bloque de segmentación para continuar con dos redes neuronales convolucionales, la primera, una red de clasificación binaria y la segunda, una red de clasificación multiclase.

Con respecto a la etapa de segmentación, una de las aportaciones de este trabajo es la utilización del algoritmo SLIC.

Cuando ingresa una imagen al prototipo, se realiza su segmentación y cada uno de esos segmentos es analizado por la red de clasificación binaria, para la detección o no de un señalamiento. El sistema fue diseñado para que, si la red de clasificación binaria al analizar cada segmento detecta en alguno de ellos un señalamiento, ya no continúe con la búsqueda de más señalamientos en otros segmentos. Eso limita al sistema a reconocer un solo señalamiento por imagen, pero esto podría ser modificado para localizar múltiples señalamientos en una misma imagen de entrada al sistema.

En el esquema actual (detección de un solo señalamiento por imagen), el tiempo de procesamiento de la red de clasificación binaria depende de la posición en que se encuentra el segmento con señalamiento dentro de la imagen original, si es de los primeros segmentos en ingresar, el tiempo de procesamiento en esta etapa durará menos. El peor escenario es una imagen donde no hay señalamiento o no lo reconozca, porque la red de clasificación binaria analizará todos los segmentos antes de pasar a la siguiente etapa. En caso de no reconocer algún segmento con señalamiento, la red de clasificación multiclase no se activa.

Se presentó el desempeño de las redes neuronales convolucionales desarrolladas. Para el caso de la red neuronal de clasificación binaria, se tienen datos de entrenamiento balanceados (número similar de datos por clase). Sin embargo, sería recomendable balancear la base de datos correspondiente a la red neuronal de clasificación multiclase. Aumentar la base de datos para mejorar la eficiencia de las redes neuronales, hacer un análisis con el uso de otras métricas de desempeño son tareas por realizar.

Se realizó una comparativa con otras arquitecturas de redes neuronales convolucionales del estado del arte, para el caso de la clasificación binaria, los valores de eficiencia arrojados por las redes del estado del arte son deficientes, lo que haría necesario un análisis más exhaustivo de ese comportamiento. Con respecto a la comparativa de la red de clasificación multiclase propuesta versus las arquitecturas del estado del arte seleccionadas, se observó exactitudes similares, siendo una ventaja para la arquitectura propuesta el número total de parámetros que utiliza.

Con respecto a la implementación del prototipo en el hardware seleccionado, a pesar de las limitaciones de la placa seleccionada, fue posible integrar todo el sistema, distribuyendo parte de las tareas del GPS en el Arduino, no obstante, la mayor parte de las funciones fueron realizadas por la Rasperry Pi, donde se procesaron las imágenes, se administró la base de datos (buscando señalamientos cercanos y actualizando los reconocidos) y se reprodujeron los mensajes de audio. Sin embargo, los tiempos obtenidos de procesamiento no corresponden a un sistema en tiempo real (se obtuvieron tiempos de procesamiento de 3-5 minutos, por lo que el sistema es inoperable. Una de las posibles opciones de mejora de hardware que no implica una sustitución de total es utilizar una tarjeta Raspberri Pi en unión con el dispositivo Movidius de Intel, que es una unidad de

procesamiento para visión (VPU) en formato USB que ofrece un acceso fácil a las capacidades de aprendizaje profundo de alto rendimiento y bajo consumo para aplicaciones IoT integradas. Lo que permitiría mejorar los tiempos de respuesta del sistema.

Con respecto al sistema de ubicación GPS, la selección y reproducción de audio, no se tuvieron dificultades, sin embargo, es importante analizar si la propuesta planteada de que el sistema puede acceder a señalamientos almacenados previamente de sitios donde ya estuvo el usuario es realmente útil, porque puede suceder que el señalamiento sea removido y las condiciones de ese entorno cambien, provocando un accidente al usuario.

Por último, será pertinente una evaluación de la forma en que se le haga llegar al usuario la información procesada por el prototipo, es decir, si puede adaptarse al uso de auriculares o manos libres, sin que se vea disminuido su sentido del oído.

7. Referencias.-

[1] Organización Mundial de la Salud. "Temas de salud. Discapacidades". Available: https://www.who.int/topics/disabilities/es/ [Accessed: 20- octubre- 2020].

[2] Instituto Nacional de Estadística y Geografía. "Estadísticas a propósito del día internacional de las personas con discapacidad (3 de diciembre). Datos nacionales". Comunicado de prensa núm. 638/19. 2-diciembre-2019.

[3] Instituto Nacional de Estadística y Geografía. "Clasificación de tipo de discapacidad-Histórica". Available: https://www.inegi.org.mx/contenidos/clasificadoresycatalogos/doc/clasificacion_de_tipo_de_discapacidad.pdf [Accesed: 20-octubre-2020].

[4] E A. Lafuente de Frutos, Educación inclusiva. Personas con discapacidad visual, 1º ed. Madrid: ITE, 2011, pp. 4-5.

[5] Unión latinoamericana de ciegos. "La discapacidad visual y las tecnologías de la información y la comunicación". Available: http://www.ulacdigital.org/wp-content/uploads/2020/01/La-Discapacidad-Visual-y-las-Tecnolog%C3%ADas-de-la-Informaci%C3%B3n-y-la-Comunicaci%C3%B3n-1-1.pdf [Accesed: 20- octubre- 2020].

[6] L. Nieto Riveiro y J. Muñoz Sevilla, "Aplicación de las tecnologías de la información y las comunicaciones en la vida diaria de las personas con discapacidad", 1º ed. A Coruña: Universidade da Coruña, Servizo de Publicacións, 2012, pp. 303-304.

[7] D. Gbenga, A. Shani , A. Adekunle. "Smart Walking Stick for Visually Impaired People Using Ultrasonic Sensors and Arduino". *International Journal of Engineering and Technology*, *9*(5), 2017, pp. 3435–3447.

[8] S. Mohapatra, S. Rout, V. Tripathi, T. Saxena and Y. Karuna, "Smart Walking Stick for Blind Integrated with SOS Navigation System," *2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI)*, Tirunelveli, 2018, pp. 441-447

[9] S. Wang and Y. Tian, Camera-Based Signage Detection and Recognition for Blind Persons, ICCHP (Computers Helping People with Special Needs), 2012.

[10] Tian, Y., Yang, X., Yi, C. et al. "Toward a computer vision-based wayfinding aid for blind persons to access unfamiliar indoor environments". Machine Vision and Applications 24, pp. 521–535, 2013.

[11] D. Kunene y H. Vadapalli, "Indoor Sign Recognition for the Blind", SAICSIT (Annual Conference of the South African Institute of Computer Scientists and Information Technologists), Johannesburgo, Sudáfrica, 2016.

[12] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning. MIT Press, 2016. www.deeplearningbook.org

[13] F. Chollet. Deep Learning with Python. Manning Publication Co, 2017.

[14] M. Afif, R. ayachi, Y. Said, E. Pissaloux and M. Atri, "Recognizing signs and doors for Indoor Wayfinding for Blind and Visually Impaired Persons", *2020 5th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*, Sousse, Tunisia, 2020, pp. 1-4.

[15] F. Zanetti, "Convolutional Networks for Traffic Sign Classification". Tesis de Maestría. Department of Signal and Systems. Chalmers University of Technology, Göteborg, Suecia. Pp. 26, 2016.

[16] Madan, Rishabh, Deepank Agrawal, S. Kowshik, Harsh Maheshwari, S. Agarwal and D. Chakravarty. "Traffic Sign Classification using Hybrid HOG-SURF Features and Convolutional Neural Networks." *ICPRAM*, 2019.

[17] Saleh, Shadi & Saleh, Hadi & Nazari, Mohammad & Hardt, Wolfram, "Outdoor Navigation for Visually Impaired based on Deep Learning". Actual Problems of System and Software Engineering (APSSE 2019), 2019.

[18] Comité Internacional Pro Ciegos I. A. P., CDMX. Available: http://lugaresaccesibles.com/lugar/comite-internacional-pro-ciegos-iap-cdmx. [Accesed: 15-octubre-2020].

[19] Información oficial del Gobierno de los Estados Unidos relativa al Sistema de Posicionamiento Global y temas afines. Sistema de posicionamiento global al servicio del mundo. Available: https://www.gps.gov/spanish.php [Accesed: 15-octubre-2020].

[20] A. Krizhevsky, I. Sutskever, G. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks". Neural Information Processing Systems. 25. 10, 2012.

[21] K.Symonian, A. Zisserman, "Very deep convolutional networks for large-scale image recognition", 2015.

[22] Vezhnevets, Konouchine , "GrowCut" - Interactive Multi-Label N-D Image Segmentation By Cellular Automata, 2005.

[23] N. Dalal, B. Triggs, "Histograms of oriented gradients for human detection". In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, volume 1, pages 886– 893. IEEE.

[24] H. Bay, T. Tuytelaars, L.Van Gool, "Surf: Speeded up robust features". In Leonardis, A., Bischof, H., and Pinz, A., editors, Computer Vision – ECCV 2006, pages 404–417, Berlin, Heidelberg. Springer Berlin Heidelberg, 2006.

[25] Secretaría de Comunicaciones y Transportes, Manual de Señalización Vial y Dispositivos de Seguridad, México, 2014, pp. 9.

[26] M. Hart, "TinyGPS++ | Arduiniana", Arduiniana.org, 2014. [Online]. Available: http://arduiniana.org/libraries/tinygpsplus/. [Accessed: 19- mayo- 2019].

[27] Stutz D., Hermans A., Leibe B. (2017). *Superpixels: An Evaluation of the State-of-the-Art*. Computer Vision and Image Understanding. doi: 10.1016/j.cviu.2017.03.007.

[28] Achanta R., Shaji A., Smith K., Lucchi, A., Fua, P., Susstrunk. S. (2012). *SLIC superpixels compared to state-of-the-art superpixel methods*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 34 (11): 2274–2281.

[29] Ren X., Malik J. (2003). *Learning a classification model for segmentation*. International Conference on Computer Vision. Vol. 1: 10–17.

[30] Achanta R., Shaji A., Smith K., Lucchi A., Fua P., Susstrunk S. (2010). SLIC Superpixels. EPFL Technical Report 149300

[31] A. Hernández, Y. González, A. Morales (2017). Implementación de algoritmo de superpixeles para la segmentación de imágenes a color. Boletín UPIITA No. 61. Available: http://www.boletin.upiita.ipn.mx/index.php/component/content/article/9-articles/23-numeros-anteriores-cyt [Accessed: 27- junio- 2020].

[32] Conversor texto-voz, Octubre 3, 2017, Available: https://es.wikipedia.org/wiki/Conversor-texto-voz.

[33] Layer weight initializers. Usage of initializers. Available: https://keras.io/api/layers/initializers/ [Accessed: 25-octubre-2020].

[34] Evaluación de modelos de clasificación: Matriz de confusión y curva ROC. http://ericmelillanca.cl/content/evaluaci-n-modelos-clasificaci-n-matriz-confusi-n-y-curva-roc [Accesed: 20-octubr-2020].

[35] T. Fawcett, "An introduction to ROC analysis". Pattern Recognition Letters, 27, 861-874, 2006.

[36] B. Zoph, V. Vasudevan, J. Shlens and Q. V. Le, "Learning Transferable Architectures for Scalable Image Recognition," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, 2018, pp. 8697-8710, doi: 10.1109/CVPR.2018.00907.

[37] G. Huang, Z. Liu, L. Van Der Maaten and K. Q. Weinberger, "Densely Connected Convolutional Networks," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, 2017, pp. 2261-2269, doi: 10.1109/CVPR.2017.243.

[38] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.