VisApp: Aplicacion de Android con Deteccion de Objetos para la Asistencia de Personas Invidentes

Ing. Alan Iván Hernández Holguín[1], Dr. Luis Carlos Méndez-González[2], Dr. Luis Alberto Rodríguez-Picón[3], Dr. Iván Juan Carlos Pérez Olguín[4]

***Resumen*—**En el presente trabajo de investigación se abordó el diseño de una aplicación para la asistencia de personas invidentes basado en algoritmos de machine learning. Incluye un sistema de detección de objetos, para el que se utilizó el algoritmo YOLO v4 entrenado con 9 clases de objetos correspondientes a artículos del hogar, y provenientes del dataset Open Images v4; soporte para comandos de voz, para proporcionar una interfaz de manos libres, y guía por voz, que retroalimenta al usuario con el estado de las consultas y de los resultados obtenidos de la detección. El sistema consiste en una aplicación de Android que incorpora los motores de reconocimiento y síntesis de voz de Google y se conecta a un entorno en la nube, para ejecutar un algoritmo de consulta y mostrar los resultados través de una aplicación Web en la interfaz de la aplicación. En cuanto a resultados, se obtuvo una aplicación de Android con un algoritmo de detección de objetos que obtuvo una precisión del 60 % y un tiempo de procesa miento de 6.7 s en promedio para cada imagen.

***Palabras clave*—**Visión por computadora, Aprendizaje automático, Detección de objetos, aplicación de Android, Persona invidente

Introducción

Con el paso del tiempo, las tecnologías que nos han rodean se han hecho cada vez más sofisticadas y han tratado de resolver uno de los retos primordiales del ser humano: la supervivencia. Esto ha dado lugar a que, en tiempos recientes, la tecnología adquiriera el papel de convertirse en una herramienta que facilitara la vida del ser humano, proporcionando medios que reduzcan el trabajo de las personas y produzcan comodidad.

Lo anterior supone otro reto para la tecnología, que consiste en vencer las dificultades que ocasionan las discapacidades; como en el caso de las visuales, las cuales pueden provocar que las personas que las padecen tengan que depender de otras personas o de herramientas que faciliten sus actividades cotidianas, como bastones y sistemas braille.

Dentro del área de los dispositivos y sistemas de asistencia, se han diseñado técnicas de reconocimiento de colores basadas en el estándar RGB, con integraciones de interfaces Text-To-Speech, en sistemas de identificación de colores [1]; se han adaptado herramientas de accesibilidad para personas invidentes por medio de sistemas que utilizan retorno de audio para informar al usuario de la forma o el color de un determinado objeto [2]; se han desarrollado sistemas de reconocimiento y localización de objetos en el entorno de una casa con soporte para internet de las cosas y servicios en la nube [3]; se han implementado técnicas de análisis de patrones para retroalimentar sistemas de asistencia inteligentes para monitorear y detectar particularidades en las rutinas diarias de un usuario [4]; y se han creado dispositivos que integran sistemas de reconocimiento facial y detección de objetos para proporcionar herramientas de asistencia visual para personas invidentes [5].

Para abordar tal problemática, en este proyecto de investigación se planteó el diseño de una aplicación de Android para asistir a las personas invidentes, la cual integra las tecnologías de reconocimiento de voz, síntesis de voz, cómputo en la nube y visión por computadora enfocada en detección de objetos.

Metodología

Implementación propuesta

El sistema propuesto consiste en una aplicación de Android implementada en un teléfono inteligente con acceso a cámara, micrófono e internet. A través de la cámara, el sistema puede capturar imágenes que serán procesadas por un algoritmo de detección de objetos y se generarán respuestas de audio con los resultados obtenidos. La aplicación se diseñó con soporte para comandos de voz, por lo que no se requiere utilizar una interfaz gráfica para controlar la aplicación. El algoritmo de detección fue implementado en una aplicación web en la nube, la cual se conecta a la aplicación de Android al momento de realizar el procesamiento de imágenes. Además, se integró una base de datos

[1] El Ing. Alan Iván Hernández Holguín es estudiante de Maestría en Tecnología del Instituto de Ingeniería y Tecnología de la Universidad Autónoma de Ciudad Juárez, Chihuahua, México. al216737@alumnos.uacj.mx
[2] El Dr. Luis Carlos Méndez González es Profesor del Instituto de Ingeniería y Tecnología de la Universidad Autónoma de Ciudad Juárez, Chihuahua, México. luis.mendez@uacj.mx
[3] El Dr. Luis Alberto Rodríguez Picón es Profesor del Instituto de Ingeniería y Tecnología de la Universidad Autónoma de Ciudad Juárez, Chihuahua, México. luis.picon@uacj.mx
[4] El Dr. Iván Juan Carlos Pérez Olguín es Profesor del Instituto de Ingeniería y Tecnología de la Universidad Autónoma de Ciudad Juárez, Chihuahua, México. ivan.perez@uacj.mx

que registra los resultados de las consultas. Como se muestra en la figura 1, este proceso está integrado por cuatro módulos:

- **Recepción de datos.** Se utiliza el micrófono y la cámara del teléfono para captar la voz del usuario e imágenes, respectivamente.
- **Aplicación de Android.** La aplicación comienza el procesamiento de la información obtenida y se conecta con los módulos de detección y consulta para proporcionarle una respuesta al usuario.
- **Detección de objetos.** Procesa las imágenes capturadas y proporciona resultados al algoritmo de consulta.
- **Algoritmo de consulta.** Administra las consultas realizadas por el usuario y lo retroalimenta con los resultados obtenidos.

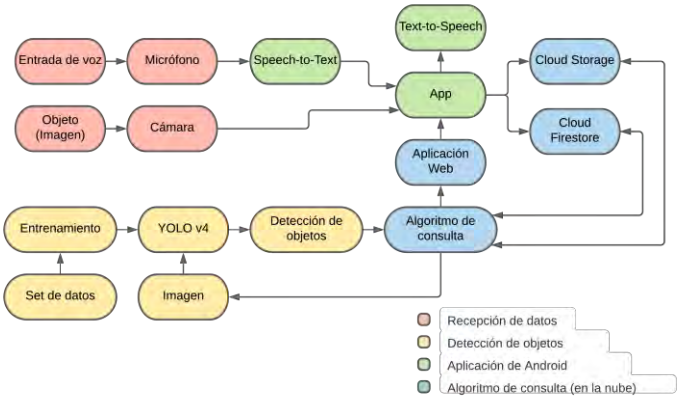

Figura 1. Proceso propuesto para la aplicación de asistencia

Software utilizado

Se utilizaron las siguientes herramientas de software para el desarrollo de la aplicación:

- Algoritmo de detección: Consiste en una versión modificada de YOLO v4 entrenada con el dataset OpenImages v4. Se desarrolló en Python en el entorno de Google Colaboratory.
- Aplicación de Android: Fue desarrollada en Java en el entorno de Android Studio.
- Base de datos: Fue creada y administrada en el entorno de Firebase utilizando los módulos Cloud Firestore y Cloud Storage.

Conjunto de datos

Se utilizó la base de datos de Google Open Images v4 [6] como fuente primaria de las imágenes que fueron utilizadas en el entrenamiento del algoritmo. Esta base de datos está compuesta por alrededor de 9 millones de imágenes que cuentan con anotaciones de etiquetas, indicaciones de relaciones entre objetos y cajas dibujadas que delimitan la localización de los objetos dentro de la imagen. De las 600 clases que conforman al conjunto de datos original, se extrajeron 9 clases (que se muestran en la tabla 1).

Clases de objetos		
ID	Clase	Imágenes por clase
0	Persona	200
1	Tijeras	200
2	Moneda	200
3	Taza	200
4	Cuchillo	200
5	Botella	200
6	Calcetín	200
7	Cuchara	200
8	Tenedor	200

Tabla 1: Clases extraídas del conjunto de datos Open Images

Entrenamiento del algoritmo

El entrenamiento se realizó ejecutando el algoritmo sobre el set de imágenes de entrenamiento utilizando los parámetros de aprendizaje. Este proceso consistió en un ciclo de procesamiento (ver figura 2) de las imágenes del conjunto de datos hasta alcanzar el valor de error mínimo. El modelo producido por el entrenamiento proporciona dos tipos de resultados, el primero consiste en etiquetas que contienen los nombres de las clases de los objetos que el algoritmo detectó en la imagen procesada, mientras que el segundo corresponde con los porcentajes de confianza asignados por el algoritmo a los resultados generados.

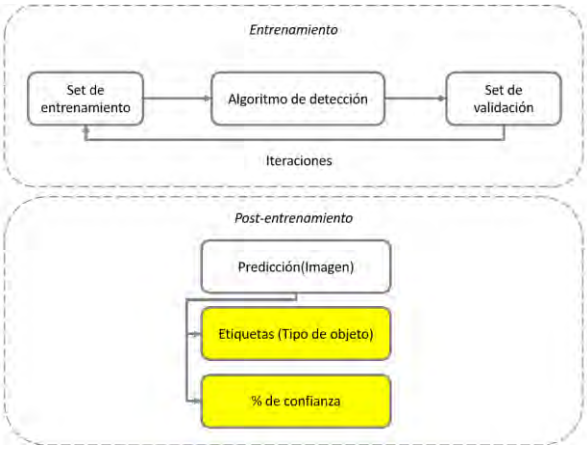

Figura 2: Proceso de entrenamiento del algoritmo de detección

Base de datos y aplicación Web

Se utilizó la plataforma Firebase de Google para crear las bases de datos donde se almacenaron los archivos de imágenes y sus registros para poder acceder a ellos en el entorno en la nube. En este caso, se utilizaron los servicios de Cloud Firestore para los registros de las variables que contienen los datos de las imágenes y Cloud Storage para almacenar los archivos de imagen. En la figura 3 se muestra el modelo de actualización de las bases de datos. Se utilizaron dos servicios de Firebase para recopilar los datos generados por la aplicación: Cloud Firestore, que almacena los datos de los procesos de detección iniciados por el usuario; y Cloud Storage, que guarda las imágenes capturadas con la cámara para que puedan ser suministradas al algoritmo de detección.

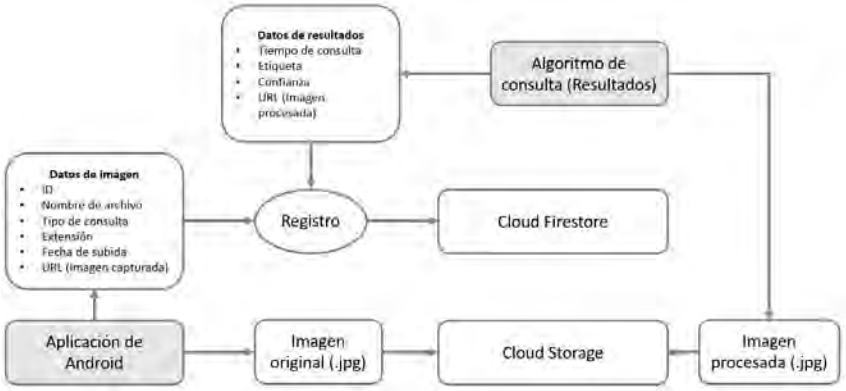

Figura 3: Proceso de actualización y consulta de la base de datos de Firebase

El algoritmo de detección se implementó como parte de una aplicación Web para reducir el consumo de recursos de la aplicación que se produciría al utilizar una versión más compacta en Android. La interfaz (mostrada en la figura 4) consiste en una ventana que muestra contenedores de imagen y texto, los cuales muestran los

resultados de la predicción realizada por el algoritmo de reconocimiento. La vista de está interfaz es cargada en la aplicación de Android por medio de una conexión a internet.

Figura 4: Aplicación Web utilizada para ejecutar la detección de objetos

Aplicación de Android

La aplicación fue diseñada para el sistema Android utilizando la plataforma de Android Studio en Java. Se utilizó la API 23 de Android (equivalente a Android 6.0) para su desarrollo. La interfaz de la aplicación se compone de dos secciones: la interfaz táctil y la interfaz por comandos de voz. El usuario es capaz de utilizar la que desee o incluso puede combinar ambas. En la figura 5 se presenta el modelo de la Aplicación de Android y las interacciones que puede activar el usuario.

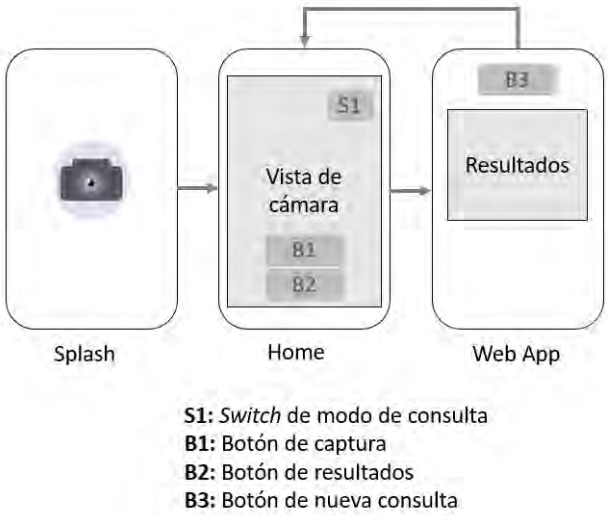

Figura 5: Diagrama de la interfaz de la aplicación de Android

Resultados

Análisis de resultados de detección

Para el modo de detección (la respuesta consiste en un conteo de los objetos detectados en la imagen, por clase), se realizaron 6 pruebas con grupos de objetos pertenecientes a las clases con las que se entrenó el algoritmo de detección. En la figura 6 se muestran las imágenes capturadas con la aplicación. En la tabla 3 se presentan los

datos obtenidos de las consultas en el modo de detección, que incluyen el tiempo de detección y el conteo de los objetos por cada imagen.

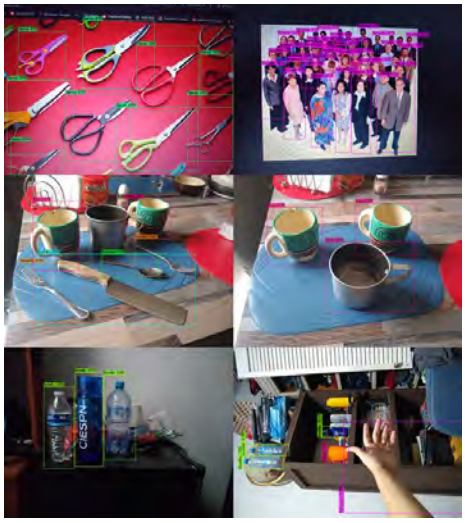

Figura 6: Resultados de detección múltiple obtenidos con la aplicación

Resultados de detección de objetos				
# prueba	Resultado general	% precisión	Tipo de resultados	Tiempo de detección
1	Tijeras: 9	93.67	6 VP, 3 FN	6.47s
2	Persona: 34	82.59	34 VP, 12 FN	6.61s
3	Cuchara: 1, Cuchillo: 1, Taza: 2, Tenedor: 2	85.67	6 VP, 1 FN	6.34s
4	Taza: 3	98	3 VP	6.16s
5	Botella: 3	96.67	3 VP	6.51s
6	Botella: 3, Persona: 1, Taza: 1	89.9	5 VP, 1 FN	6.65s

Tabla 1: Resultados de métricas de detección

De acuerdo con los resultados mostrados en la tabla 3, el algoritmo de detección implementado en la aplicación resulto ser eficaz para reconocer múltiples objetos. Además, que los valores de precisión estuvieron encima del umbral del 80 %, por lo que se considera que el entrenamiento fue adecuado. Por otro lado, debido a que el algoritmo de detección no se ejecuta localmente y necesita de una conexión a internet para ejecutarse, se contó con un tiempo de ejecución de hasta 6.65 s, que incluye el tiempo que tarda la conexión en responder y proporcionar los datos solicitados. Lo cual podría verse afectado por la rapidez de la conexión y el tipo de red en la que se encuentra el teléfono utilizado.

Conclusiones

El sistema desarrollado en este proyecto de investigación permitió combinar las tecnologías de visión por computadora, machine learning y aplicaciones web con el entorno de desarrollo de Android para generar un sistema de identificación de objetos para personas con problemas visuales.

Uno de los principales beneficios del sistema es que puede ser utilizado de forma extensiva en dispositivos Android compatibles, lo que supone una mejora de disponibilidad con respecto a otras investigaciones en la que se utilizaban microcontroladores o microcomputadoras, lo que a su vez permite una mayor flexibilidad para desplegar nuevas funciones en el sistema desarrollado en esta investigación. Además, como el algoritmo de detección se

ejecuta en un entorno en la nube, se facilita su actualización y permite cargar algoritmos y bases de datos para diferentes fines (como el reconocimiento facial o extender la cantidad de objetos que pueden ser identificados), de forma independiente al código de la aplicación.

Dentro del ámbito de la accesibilidad, el sistema facilita este aspecto para las personas con discapacidad visual al ofrecer soporte para comandos de voz que permiten controlar por completo las actividades desarrolladas por la aplicación.

Limitaciones

Este trabajo estuvo limitado por los siguientes aspectos:

- El algoritmo de detección y consulta se ejecutan en un entorno en la nube, por lo que se requiere de una conexión a Internet.
- El tiempo de respuesta de los servicios que utilizan la app para realizar las detecciones es de alrededor de 10 s
- Las clases de objetos que pueden ser identificadas por el algoritmo están limitadas a 9 tipos diferentes
- La aplicación solo está disponible en la plataforma Android
- Se utilizó la versión 4 del algoritmo YOLO, por lo que podrían obtenerse resultados diferentes con una versión más actualizada.

Recomendaciones

El sistema desarrollado en esta investigación puede sentar las bases para investigaciones futuras, relacionadas con el desarrollo de sistemas de accesibilidad enfocados en la identificación de objetos. Se podría desarrollador un sistema basado en detección de objetos con retroalimentación en tiempo real, que busque objetos capturados en vídeo en tiempo real de acuerdo con las peticiones del usuario (por ejemplo, que el usuario proporcione el nombre de un tipo de objeto; este enfoque su cámara en distintos puntos de su hogar mientras captura vídeo y si se encuentra el objeto, que se le proporcione una respuesta al usuario). Otro enfoque sería utilizar un sistema de cámaras que capturen un entorno controlado (como el interior de una casa) y utilicen técnicas de detección y correlación entre objetos para crear una herramienta de asistencia que pueda notificar al usuario de objetos olvidados o incluso pueda ser configurado como un sistema de vigilancia inteligente. Además, podrían implementarse otros enfoques de visión por computadora, como reconocimiento facial o de características. En general, se puede trabajar en los siguientes aspectos:

- Tipo de algoritmo de detección
- Integración de funciones en tiempo real
- Implementación de reconocimiento facial
- Diseño de sistemas de vigilancia o de detección de patrones en objetos

Referencias

[1] Mungkaruna, P. Piyawongwisal, K. Ropkhop, and U. Hatthasin, “The talking color identifying device for the visually impaired,” 2016 13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications, and Information Technology, ECTI-CON 2016, 2016.

[2] S. Jamuni and S. Borkar, “Colour and shape identification for assisting visually impaired person,” Proceedings of the International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), I-SMAC 2018, pp. 415–418, 2019.

[3] K. Srinivasan and V. R. Azhaguramyaa, “Internet of Things (IoT) based Object Recognition Technologies,” Proceedings of the Third International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC 2019), pp. 216–220, 2019.

[4] R. Phadnis, J. Mishra, and S. Bendale, “Objects Talk - Object Detection and Pattern Tracking Using TensorFlow,” Proceedings of the International Conference on Inventive Communication and Computational Technologies, ICICCT 2018, pp. 1216–1219, 2018.

[5] Joe Louis Paul, S. Sasirekha, S. Mohanavalli, C. Jayashree, P. Moohana Priya, and K. Monika, “Smart Eye for Visually Impaired-An aid to help the blind people,” ICCIDS 2019 - 2nd International Conference on Computational Intelligence in Data Science, Proceedings, 2019

[6] Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov. “The open images dataset v4,” International Journal of Computer Vision, vol. 128, no. 7, 2020. [Online]. Available: http://dx.doi.org/10.1007/s11263-020-01316-z