

Detección Automática de Objetos como asistencia a Personas Invidentes

» López De Luise Daniela

CAETI – Universidad Abierta Interamericana – Facultad de Tecnología Informática
Av. Montes de Oca 745, Ciudad de Buenos Aires, Argentina.

CI2S Labs

Pringles 50, Ciudad de Buenos Aires, Argentina.

Universidad Cuenca del Plata, Corrientes, Argentina – Facultad de Ingeniería y Tecnología
Lavalle 50, Corrientes, Argentina.

daniela_ldl@ieee.org

Park Jin Sung

CI2S Labs

Pringles 50, Ciudad de Buenos Aires, Argentina.

zeroalpha2000@gmail.com

Hoferek Silvia

Universidad Cuenca del Plata, Corrientes, Argentina – Facultad de Ingeniería y Tecnología
Lavalle 50, Corrientes, Argentina.

hofereksilviarosana_for@ucp.edu.ar

Universidad Cuenca del Plata, Formosa, Argentina – Facultad de Ingeniería y Tecnología-
Maipú 4625, Formosa, Argentina.

srhoferek@gmail.com

Avila Lautaro Nicolás

Universidad Cuenca del Plata, Corrientes, Argentina – Facultad de Ingeniería y Tecnología
Lavalle 50, Corrientes, Argentina.

lautyx027@gmail.com

Universidad Cuenca del Plata, Formosa, Argentina – Facultad de Ingeniería y Tecnología-
Maipú 4625, Formosa, Argentina.

lautyx027@gmail.com

Benitez Micaela Antonella

Universidad Cuenca del Plata, Corrientes, Argentina – Facultad de Ingeniería y Tecnología
Lavalle 50, Formosa, Argentina.

Universidad Cuenca del Plata, Formosa, Argentina – Facultad de Ingeniería y Tecnología-
Maipú 4625, Formosa, Argentina.

benitezmicaelaantonela@gmail.com

Bordon Sbardella Felix Raul

Universidad Cuenca del Plata, Corrientes, Argentina – Facultad de Ingeniería y Tecnología
Lavalle 50, Formosa, Argentina.

Universidad Cuenca del Plata, Formosa, Argentina – Facultad de Ingeniería y Tecnología-
Maipú 4625, Formosa, Argentina.

raulbordon250@gmail.com

Fantín Rodrigo Iván

Universidad Cuenca del Plata, Corrientes, Argentina – Facultad de Ingeniería y Tecnología
Lavalle 50, Formosa, Argentina.

Universidad Cuenca del Plata, Formosa, Argentina – Facultad de Ingeniería y Tecnología-
Maipú 4625, Formosa, Argentina.

rodrignr4@gmail.com

Machado Gastón Emmanuel

Universidad Cuenca del Plata, Corrientes, Argentina – Facultad de Ingeniería y Tecnología
Lavalle 50, Formosa, Argentina.

Universidad Cuenca del Plata, Formosa, Argentina – Facultad de Ingeniería y Tecnología-
Maipú 4625, Formosa, Argentina.

gastonmachado44@gmail.com

Mencia Aramis Oscar

Universidad Cuenca del Plata, Corrientes, Argentina – Facultad de Ingeniería y Tecnología
Lavalle 50, Formosa, Argentina.

Universidad Cuenca del Plata, Formosa, Argentina – Facultad de Ingeniería y Tecnología-
Maipú 4625, Formosa, Argentina.

aramismencia@gmail.com

Ríos Anahí Ailén

Universidad Cuenca del Plata, Corrientes, Argentina – Facultad de Ingeniería y Tecnología
Lavalle 50, Formosa, Argentina.

Universidad Cuenca del Plata, Formosa, Argentina – Facultad de Ingeniería y Tecnología-
Maipú 4625, Formosa, Argentina.

anahiriosailen@gmail.com

Rios Emiliano Luis

Universidad Cuenca del Plata, Corrientes, Argentina – Facultad de Ingeniería y Tecnología
Lavalle 50, Formosa, Argentina.

Universidad Cuenca del Plata, Formosa, Argentina – Facultad de Ingeniería y Tecnología-
Maipú 4625, Formosa, Argentina.

mcwiths@gmail.com

Riveros Nahuel Edgardo

Universidad Cuenca del Plata, Corrientes, Argentina – Facultad de Ingeniería y Tecnología
Lavalle 50, Formosa, Argentina.

Universidad Cuenca del Plata, Formosa, Argentina – Facultad de Ingeniería y Tecnología-
Maipú 4625, Formosa, Argentina.

Nahuel42425@gmail.com

Resumen

El objetivo de este artículo es presentar un prototipo de asistencia a personas ciegas. Utiliza los dispositivos de un teléfono móvil estándar y su diseño minimiza el requerimiento computacional. El trabajo aquí presentado es el resultado de una serie de actividades que abarcan la recolección de videos, el pre-procesamiento para reducir el streaming, la segmentación de imágenes y el procesamiento con Minería de datos. Los modelos derivados se emplean en la detección de objetos, perfilado del comportamiento del individuo, detección y evaluación de movimientos en el entorno y la interpretación de eventos relevantes en tiempo real a fin de activar alarmas específicamente diseñadas para asistir al invidente. Asimismo se introduce el prototipo que implementa a través de estos modelos las técnicas de procesamiento de imágenes, detección de áreas y clasificación de objetos depurados a fin de inferir si existe en el entorno algún tipo de riesgo. Con este dispositivo se pretende asistir a las persona ciegas determinando la naturaleza de los objetos y circunstancias del entorno, con el fin de advertir con anticipación eventualidades relevantes que merezcan especial atención. El alcance del trabajo comprende antecedentes del sector, primeros hallazgos, variables y características del prototipo derivado, peculiaridades detectadas en la interacción con personas ciegas y la funcionalidad global del dispositivo, así como un pequeño análisis de los resultados.

PALABRAS CLAVE: ASISTENCIA AL INVIDENTE, PROCESAMIENTO DE VIDEO, DETECCIÓN DE OBJETOS, MINERÍA DE DATOS, PREDICCIÓN DE TRAYECTORIAS.

Automatic Object Detection as Assistance to Blind People

Abstract

The aim of this article is to present a prototype of assistance to blind people. It uses the devices of a standard mobile phone and its design minimizes the computational requirement. The work presented here is the result of a series of activities covering video collection, pre-processing to reduce streaming, image segmentation and processing with Data Mining. The derived models are used in the detection of objects, profiling of the behavior of the individual, detection and evaluation of movements in the environment and the interpretation of relevant events in real time in order to activate alarms specifically designed to assist the blind. Likewise, the prototype that implements through these models the techniques of image processing, detection of areas and classification of purified objects is introduced in order to infer if there is any type of risk in the environment. This device is intended to assist blind people by determining the nature of objects and circumstances in the environment, in order to warn in advance of relevant eventualities that deserve special attention. The scope of work includes background of the sector, first findings, variables and characteristics of the derived prototype, peculiarities detected in the interaction with blind people and the overall functionality of the device, as well as a small analysis of the results.

KEYWORDS: ASSISTANCE TO THE BLIND, VIDEO PROCESSING, OBJECT DETECTION, DATA MINING, TRAJECTORY PREDICTION.

Introducción

En la actualidad existen pocas propuestas prácticas, disponibles y efectivas para solventar los inconvenientes que genera la movilidad en exteriores por parte de personas con discapacidad visual [01][02]. Una de las soluciones típicas es el bastón, o la asistencia de terceros (sean personas o animales entrenados especialmente). Las herramientas mecánicas y tecnológicas y propuestas disponibles en el mercado que son costosas y muchas veces incómodas de utilizar. Entre otras se pueden mencionar tecnología asistiva para estudiantes [03], orientación y movilidad [04], percepción y aprendizajes, entre otras.

El enfoque del proyecto actual requiere de escasa tecnología: un teléfono celular estándar y opcionalmente algunos accesorios para mejorar la detección y predicción [05][06][07]. Es importante destacar que esta problemática afecta la calidad de vida de las personas involucradas, específicamente aquellas que no cuentan con recursos accesibles para desplazarse de manera independiente.

Comprender los riesgos e inconvenientes que sufren estas personas al desplazarse en distintos entornos requiere de la colaboración con especialistas y profesionales del entorno. Por tal motivo el presente proyecto involucra el trabajo conjunto con el Círculo de no Videntes (CINOVI) [08]. De esta manera se validan las estrategias definidas para las detecciones y reacciones a modelar. El resultado del trabajo previo realizado en el proyecto con miembros de la Biblioteca Argentina para Ciegos (BAC) [09], del centro CAETI de UAI [10] y del equipo actual junto al CINOVI se refleja en un prototipo denominado HOLOTECH. Se trata de un sistema del tipo denominado mobile (sobre dispositivos móviles), que funciona con dispositivos móviles de gama media con Android, y cuyo objetivo es asistir a personas con problemas severos de visión emitiendo señales sonoras ante determinados obstáculos y situaciones.

El resto de este paper se organiza de la siguiente manera: sección II describe la metodología y materiales, sección III las pruebas de campo y secciones IV y V los resultados del modelo obtenido con sus estadísticas y conclusiones respectivamente.

II. Metodología y Materiales

El software y hardware del prototipo respeta las restricciones y requerimientos obtenidos de las primeras etapas del proyecto con BAC. La metodología de trabajo aplicada responde a los siguientes pasos:

- a) Generación de datos de test: Como primera etapa para la prueba de campo, la base de datos se genera grabando videos de 1 minuto. Los equipos empleados son celulares de gama media con una cámara de video, 64 GB de memoria interna y sistema operativo Android. Las condiciones siguen los lineamientos de un usuario estándar del prototipo con impedimento visual. Entre las consideraciones se determina la altura del dispositivo en la zona del pecho, simular un desplazamiento a velocidad lenta, la existencia de obstáculos fijos y móviles y con un dispositivo celular no profesional. Los registros responden a categorías según el objetivo del aprendizaje. Las capturas se conforman de una resolución estándar que va desde 320x240 dpi a 720x480 dpi. El formato es XML [11].

b) Preprocesamiento de datos: A fin de lograr una producción sistemática el proceso preliminar se realiza con módulos específicamente desarrollados. Entre otros, los procesos en esta etapa comprenden: pasaje a blanco y negro, blur y eliminación del contorno, filtrado [12][13], reconstrucción [14].

c) Entrenamiento del modelo neuronal: Los datos limpios y clasificados se ingresan a la red, ya parametrizada. Por medio de un proceso de optimización se ajustan los metadatos hasta que se logra una tasa de error y métricas de validación (log likelihood y error cuadrático medio) aceptables.

Las herramientas de software empleadas son OpenCV [15] y la librería Pandas en un entorno de programación Python.

Para la preparación de los datos luego de evaluar las posibles opciones se opta por el lenguaje Python por su capacidad de ser multiparadigma y la adaptabilidad que posee.

En cuanto a los Frameworks se trabaja con Kivy [16], luego de evaluar también las alternativas Django [17] y Flask [18]. La elección se debe a que posee una buena integración con el ambiente móvil y presenta facilidad de uso. Para el front-end se emplean Flet [19] y Coffe [20].

Para el procesamiento de imágenes las bibliotecas principales son YOLO [21] y OpenCV, cuyo nivel de estabilidad en la comunidad permite un confiable proceso de etiquetado e integración de heurísticas de Inteligencia Artificial.

Android Studios [22] es la herramienta del proyecto para probar la compatibilidad de las versiones empleadas contra diferentes dispositivos móviles. Finalmente cabe mencionar que Buildozer [23] es el empaquetador de la APK [24] y Labelme [25] se emplea para crear nuevas etiquetas en el Log de rastreo que sirve para entrenar la red neuronal con distintos objetivos, y para generar las pruebas de campo.

III. El Prototipo y el Desarrollo de las pruebas

Las muestras de video, organizadas en carpetas, corresponden a tomas en un complejo Universitario y en la vía pública, que se organizan en carpetas. Se distinguen videos completos y recortados, tomas en exterior e interior, diurnas y nocturnas. A fin de realizar un entrenamiento aceptable los registros mayoritariamente corresponden a sectores de calles, veredas o vehículos.

Los videos completos son editados eliminando el audio, ya que para esta etapa no es relevante y puede ayudar a reducir el peso del archivo. A continuación se realiza la segmentación según los distintos ambientes que predominen y en caso de ser necesario disminuir la calidad de los clips para que sean más livianos.

Es importante destacar que los aportes del CINOVI permitieron validar la lista de obstáculos y eventos de interés para el sistema. También se incorporaron problemáticas que no se habían tenido en cuenta. En la lista de eventos y objetos a detectar y reaccionar se encuentran:

- Autos mal estacionados en las veredas o en espacios no permitidos, por ejemplo, en las bocacalles.
- Motos/bicicletas estacionadas en lugares no permitidos o circulando irregularmente.
- Portones abiertos o que abren hacia la calle.
- Ramas de plantas o algún objeto que sobresalga en la ruta de desplazamiento, a partir de la altura de las rodillas hacia arriba.
- Existencia de rampas municipales.
- Presencia de dispositivos como aire acondicionado a alturas no recomendadas, principalmente si sobresalen de la pared.
- Existencia de pozos o roturas en el suelo, con o sin vallado.
- Presencia de obstáculos móviles que podrían ser peligrosos.

En el listado se destacan los objetos u obstáculos que se encuentran por encima de la cintura, puesto que no son detectables por un bastón.

El Prototipo

El prototipo desarrollado con Kivy comienza con un avance preliminar basado en un repositorio de prueba llamado “Caffe-SSD-Object-Detection” [26], que detecta objetos por medio de la biblioteca de OpenCV y YOLO. De esta manera, la implementación se divide por etapas que gradualmente pasan de herramientas estándares al proyecto adaptado para HOLOTECH. Una vez que esta etapa funciona, el siguiente paso consta de la selección de algunas partes del proyecto para extenderlo con componentes de la biblioteca OpenCV.

Para crear prototipos de los menú y su posible visualización en un celular se emplea Flet, un Framework estable apto para este tipo de tareas [27][28]. Otro framework probado fue Android Studio, el cual fue más problemático para el acceso a la cámara, versionados que impedían las pruebas, y la integración con python que presentaba incompatibilidad. Dado el tiempo requerido para solucionar estas dificultades, se decidió descartarlo y proceder con el Buildozer.

Para poder utilizar Buildozer se necesita trabajar en un ambiente basado en Linux, por lo que se plantearon dos soluciones: Utilizar el WSL [29] de Windows o utilizar una máquina virtual. Un primer intento con WSL para la creación del apk [30], presentó problemas en la ejecución de Buildozer y la compilación de APK no pudo concretarse, por lo que se optó por otras alternativas.

Finalmente se pudo realizar la creación de la APK usando la máquina virtual. Entonces surgió otro inconveniente: no se podía entregar la apk al dispositivo. Analizando el funcionamiento del Buildozer, este genera las herramientas y utilidades para una aplicación en un ambiente Android y necesita tener conectado un dispositivo para poder entregar y cargar la apk. En la configuración de desarrollo esto no era posible, ya que la máquina no lograba reconocer el celular conectado. Finalmente esto se pudo resolver con un cambio de dispositivo.

En un principio se utilizó el sistema operativo Manjaro [31] basado en Arch Linux [32], gracias a la propuesta de uno de los integrantes, pero generó múltiples conflictos de compatibilidad con el Kivy y Buildozer por lo que se decidió emplear el sistema operativo Ubuntu [33].

Desarrollo de una prueba y armado de dato

El armado de datos, requiere que los objetos a reconocer sean detectados en las imágenes de entrenamiento. Para ello, empleando la herramienta labelme se generan etiquetas que permiten crear una identidad para los objetos de interés. Su funcionamiento consiste en la creación de una clase de objeto al que identifica porque se lo delimita previamente en forma manual. Este trabajo requiere que una a una las imágenes de entrenamiento tengan definida la forma de una cápsula que englobe la entidad de interés y los puntos más importantes que la componen.

La primera prueba tiene como objetivo la detección de puertas. Luego de compilar dos videos de prueba (puertasAdentro1.mp4 y puertasAdentro2.mp4), con 30 fotogramas por segundo, se les colocó los puntos de coordenadas y se procesó mediante YOLO. Los resultados indicaron que el sistema era incapaz de detectar más del 50 % de los casos, por lo que se dedujo que la cantidad de casos no eran suficientes como para poder tener un reconocimiento confiable. Entre otras distorsiones, generaba etiquetas para objetos o personas que no lo eran.

La segunda prueba incorpora un 40% más de imágenes. Una vez ajustado de este modo el conjunto de entrenamiento, se procedió a repetir el procesamiento con el mismo protocolo. Los resultados obtenidos sobre los objetos detectados en un trayectoria permiten identificar la ubicación del objeto y el análisis necesario para determinar el comportamiento del mismo.

startX	startY	endX	endY	objectName	assertion	timestamp
420	320	635	546	sofa	32.68%	0:00
407	318	631	546	sofa	32.67%	0:00
414	355	633	554	sofa	36.35%	0:00
396	334	637	548	sofa	47.46%	0:00
348	349	602	572	sofa	40.05%	0:00
8	645	829	1676	diningtable	34.48%	0:00
11	717	807	1644	diningtable	36.85%	0:00

Fig. 1 Estructura de datos

De los datos obtenidos es posible detectar cambios en la ubicación de los objetos según sean las distintas tomas y es posible observar la certeza de la detección del mismo objeto. A manera de ejemplo, en la Fig. 2 se observa una silla con un marco de color lima y su correspondiente etiqueta (“chair”). De manera similar, en la Fig. 3 se presenta como objeto detectado una persona, su marco en color morado y la etiqueta correspondiente (“person”). Las detecciones son estables con variaciones lentas en cada slice del video, por lo que no varían significativamente en varias tomas y, como se verá en la sección siguiente se puede inferir que el entrenamiento para el reconocimiento del mismo es confiable. Estos datos permiten evaluar el tipo de movimiento, una noción de su ubicación y trayectoria sobre el espacio. Hay que tener en cuenta en el cálculo del desplazamiento del movimiento del propio usuario, para determinar la trayectoria del objeto correctamente.



Fig. 2 Detección de una Silla



Fig.3 Deteccion de las Personas

Los modelos iniciales que se describen en la siguiente sección empleados sobre datos, permiten vislumbrar los espacios o zonas de riesgo, indispensable para que el invidente pueda tomar decisiones y evitar colisiones o accidentes. Cuando los ejes de referencia en relación a la posición del objeto se acercan o permanecen en las zonas de riesgo (como en el centro del eje X de la captura que se puede visualizar en la Fig.4) se emiten las alarmas correspondientes que el usuario haya definido de acuerdo a sus preferencias personales y necesidades especiales.

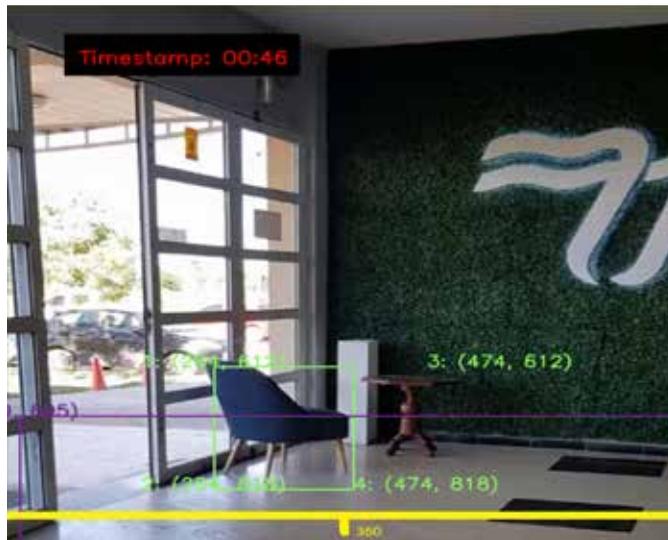


Fig.4 Detección por Coordenadas del objeto "chair" y eje X con marcado del punto medio

IV. Modelado y estadísticas

Los videos procesados por el prototipo generan un archivo de Log de 1320 registros con la registración de los resultados (match) por cada tipo de objeto hallado, y un detalle del timestamp (momento de la detección) y las coordenadas. La Fig. 5 muestra la distribución de objetos detectados en un conjunto de datos usados para la prueba.

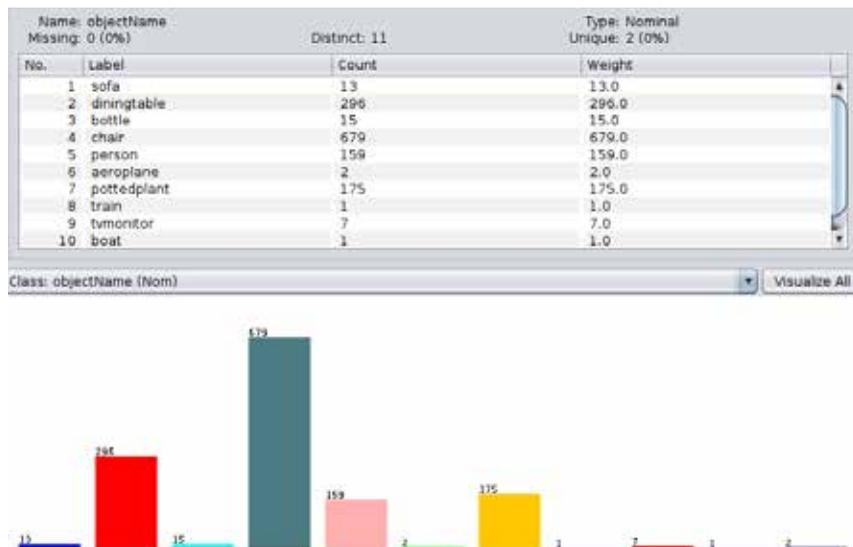


Fig. 5 Objetos del dataset

La figura presenta las categorías de objeto de la tabla como barras de colores, que se corresponden de izquierda a derecha con los objetos numerados desde el 1 en adelante. Se puede observar que la distribución mayoritaria corresponde a chair (sillas), con 679 instancias. Le siguen las mesas (diningtable) con 296 instancias, maceta con planta (pottedplant) con 175 y personas con 159 respectivamente. El resto de los objetos son minoritarios. La Fig. 6 muestra cada una de las clases de objeto en que posición relativa al borde izquierdo se los ha detectado.

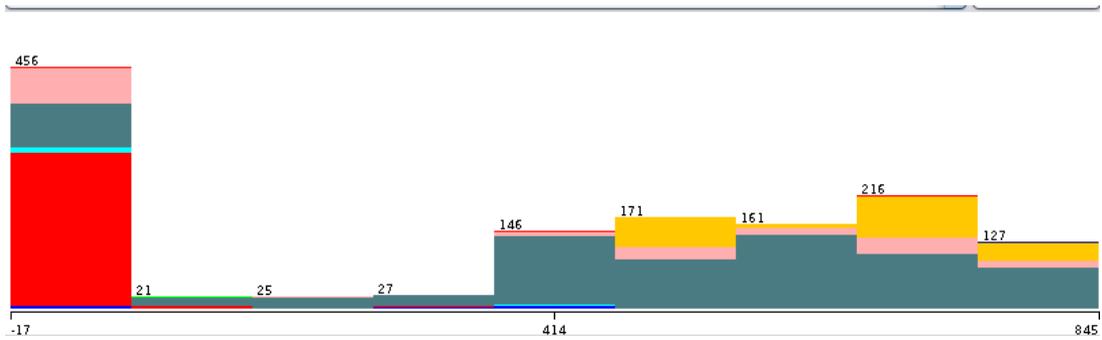


Fig. 6 Distribución de los objetos en la imagen

Se observa que las sillas y las personas pueden estar prácticamente en cualquier lugar del espacio, mientras que las macetas suelen estar del lado derecho de la imagen y las mesas del lado izquierdo. Esto puede ser un sesgo de captura de los videos. En la Fig. 7 se observa la tasa de acierto por cada tipo de objeto.

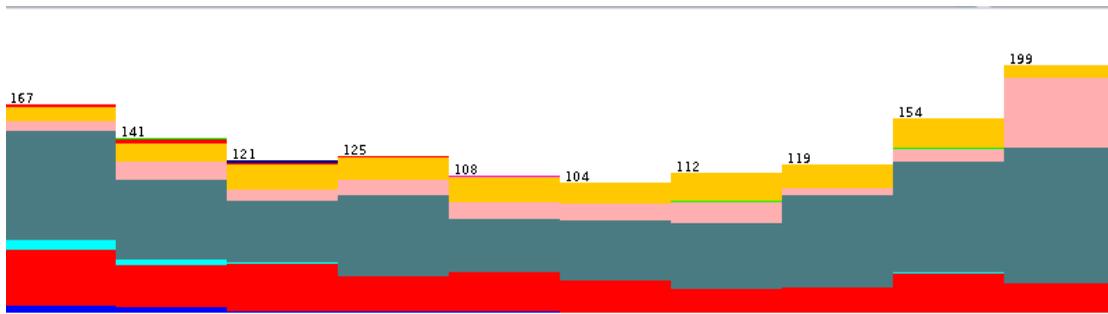


Fig. 7 Tasa de acierto en la detección por tipo de objeto

Se observa que las clases mayoritarias tienen una tasa de acierto similar. Las botellas, sofás, aviones y otras clases minoritarias tienen comportamientos de detección variada, lo que se puede deber a que la cantidad de casos son pocos y la estadística no alcanza a representar el comportamiento real del modelo. El umbral de corte podría ser de unos 30 casos, lo típicamente establecido para una muestra estadísticamente representativa, lo que debe ser confirmado en futuras pruebas con más muestras.

Para los casos en que la muestra es representativa, más del 51% (688 casos) de las veces el acierto supera el 64.91%. En el 49% de los casos (662 instancias) los aciertos son por debajo de este umbral considerado aceptable para las implementaciones prácticas realizadas con personas invidentes en las pruebas de campo.

A fin de estudiar si la productividad del prototipo tiene una tendencia respecto a la detección de objetos, se realiza un estudio de agrupaciones naturales con el heurístico Density Based Clusterer de WEKA (DBC) [34]. La fig. 8 muestra las dos agrupaciones que se encuentran, distribuidas según la relación entre tipo de objeto y la tasa de aciertos (match).

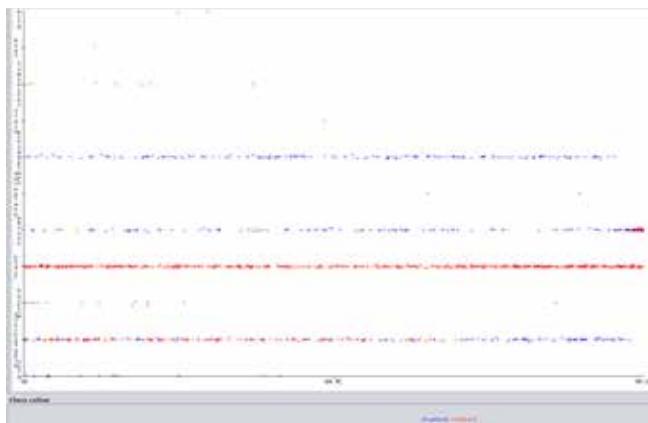


Fig. 8 Distribución de agrupaciones por tipo de objeto según la tasa de aciertos

Se puede observar en la figura a $x = \text{match}$, $Y = \text{ObjectName}$, donde los clusters se comportan distinto según se trate de ciertos objetos (Cluster 0 está en azul y Cluster 1 en rojo). Evaluando estos mismos datos con otros heurísticos como Farther First, Filtered Clusterer y Hierarchical Clusterer, todos basados en las propiedades intrínsecas y sus distancias numéricamente evaluadas, los resultados son prácticamente los mismos. De lo anterior surge que las dos agrupaciones representan comportamientos distintos.

Analizando ambos clusters se obtiene la descripción que se detallan en la Tabla 1 y 2 desde la plataforma WEKA.

Tabla 1. Características del Cluster 0

Cluster: 0 Prior probability: 0.2559	
Attribute	Normal Distribution. Mean (D)/Estimator. Counts(E) Mean
id	(D) 444.8667 StdDev = 263.6751
startX	(D) 82.7652 StdDev = 195.8292
startY	(D) 688.2696 StdDev = 216.1976
endX	(D) 880.4 StdDev = 120.5797
endY	(D) 1538.8493 StdDev = 242.8797
objectName	(E) 1 297 6 1 13 1 31 2 2 1 1 (Total = 356)
match	(D) 60.4206 StdDev = 20.3299
timestamp	(E) 14 13 16 10 11 13 13 10 10 12 11 16 5 31 1 1 13 12 24 31 17 12 17 16 7 4 6 9 9 3 1 1 1 1 2 1 1 1 1 1 8 1 1 1 1 (Total = 390)

Tabla 2. Características del Cluster 1

Cluster: 1 Prior probability: 0.7441	
Attribute	Normal Distribution. Mean (D)/Estimator. Counts(E) Mean
id	(D) 753.3294 StdDev = 394.7512
startX	(D) 490.61 StdDev = 248.7056
startY	(D) 563.407 StdDev = 175.2913
endX	(D) 697.8786 StdDev = 213.7928
endY	(D) 1538.8493 StdDev = 242.8797
objectName	(E) 1 297 6 1 13 1 31 2 2 1 1 (Total = 356)
match	(D) 67.6902 StdDev = 22.1793
timestamp	(E) 18 19 16 22 21 19 19 22 22 20 21 16 27 1 31 31 19 20 8 1 15 20 15 16 25 28 26 23 23 29 31 31 31 31 30 31 31 31 31 31 24 31 31 31 31 (Total = 1050)

Donde los centroides correspondientes son los de la tabla 3.

Tabla 3. Centroides de las agrupaciones naturales

Cluster#			
Attribute	Full Data	1	
	(1350.0)	(345.0)	(1005.0)
id	674.5	444.8667	753.3294
startX	386.383	82.7652	490.61
startY	595.3163	688.2696	563.407
endX	744.523	880.4	697.8786
endY	1099.2081	1538.8493	948.2866
objectName	chair	diningtable	chair
match	65.8324	60.4206	67.6902
timestamp	00:00	00:13	00:14

Esto indica que las clases representan dos de los objetos más frecuentes. Ambos con una tasa de aciertos superior a 60%, pero diferenciados principalmente en su tamaño. Esto indica que naturalmente no se observa el sesgo estadístico diferenciado en la calidad de detección, siendo entonces altamente probable que la capacidad de detección sea similar para todos los casos.

V. Conclusiones y trabajos futuros

Es posible administrar los datos obtenidos por medio de un dispositivo celular para identificar en tiempo real objetos de interés y su ubicación. Por ello es dable esperar que se pueda hacer un análisis de riesgo de colisión y tener un entendimiento del movimiento y trayectoria de ciertos objetos. La frecuencia para obtener y analizar esta información a tiempo real es adecuada en dispositivos móviles de gama media, con un muestreo mínimo de 30 fotogramas por segundos se puede concluir que es adecuado. Por otra parte, para reducir el tiempo de cómputo es recomendable tener pre-configurado las características del usuario como su velocidad y características físicas para determinar su zona de riesgo. La incorporación de estas configuraciones no son parte del trabajo actual y son parte del trabajo futuro.

De lo observado en la sección IV se puede concluir que el conjunto de datos de prueba no está balanceado, por lo que el comportamiento no es el mismo para los casos en que se dispone de suficiente cantidad de muestreos. En los casos mayoritarios el modelo presenta un comportamiento confiable y uniforme. En los casos minoritarios los comportamientos son diversos por falta de casos suficientes para una estadística confiable. El umbral de diferenciación en este estudio podría estar dado por el de una muestra representativa y su valor preciso debe ser confirmado en futuros estudios estadísticos.

A futuro también se deberán incorporar más objetos, categorizar su comportamiento e incluir el análisis de riesgo y asimilar con el sistema de sonido, evaluar el impacto de la extensión del sistema de alarmas actual (más información en [05]), para avisar al usuario sin confundirle pero eficazmente. Además queda por agregar mayor cantidad de variables para una mayor descripción del entorno (indoor u outdoor) y validar el uso de otros dispositivos como el GPS para identificar su contexto.

Referencias

- » [01] J. Evangeline, "Guide Systems for the Blind Pedestrian Positioning and Artificial Vision", *IJSET - International Journal of Innovative Science, Engineering & Technology*, Vol. 1 Issue 3. 2014.
- » [02] R. Velázquez, "Wearable Assistive Devices for the Blind". Chapter 17 in A. Lay-Ekuakille & S.C. Mukhopadhyay (Eds.), *Wearable and Autonomous Biomedical Devices and Systems for Smart Environment: Issues and Characterization*, LNEE 75, Springer, pp 331-349. 2010.
- » [03] O. Lahav, D.W. Schloerb. "Virtual Environments for People Who Are Visually Impaired Integrated into an Orientation and Mobility Program". *American J. of Visual Impairment*. 2015.
- » [04] S.M. Kelly, P.M. Ajuwon, K.E. Wolffe, "The Recreation and Leisure Pursuits of Employed Adults with Visual Impairments in Nigeria: Part1". *American J. of Visual Impairment*. 2014.
- » [05] J. S. Park, D. López De Luise, D. J. Hemanth & J. Pérez. *Environment Description for Blind People. International Workshop Soft Computing Applications*. Springer. 2018.
- » [06] H. Weiming, X. Xiao, D. Xie, T. Tan, and S. Maybank. Traffic accident prediction using 3-D model-based vehicle tracking. *IEEE transactions on vehicular technology* 53, no. 3 pp. 677-694. 2004.
- » [07] P. Soo-Chang, W. Kuo, and W. Huang. "Tracking moving objects in image sequences using 1-D trajectory filter." *IEEE Signal Processing Letters* 13, no. 1, pp. 13-16. 2006.
- » [08] CINOVI. Asociación sin fines de lucro. Formosa, Argentina. Sitio www.cinovi.org.ar. 2023.
- » [09] Biblioteca Argentina para Ciegos. Asociación sin fines de Lucro. Ciudad Autónoma de Buenos Aires, Argentina. Sitio bac.org.ar. 2023.
- » [10] CAETI. Centro de Altos Estudios en Tecnologías Informáticas perteneciente a la Universidad Abierta Interamericana. Sitio caeti.uai.edu.ar. 2023.
- » [11] Documentación de Python - 3.11.5 XML. Sitio: docs.python.org. 2023.
- » [12] E. Dubois, and S. Shakeri. Noise reduction in image sequences using motion-compensated temporal filtering. *IEEE transactions on communications* 32, no. 7 pp. 826-831. 1984.
- » [13] M. Sezan, K. Ibrahim, K. Mehmet, and V. Fogel. Temporally adaptive filtering of noisy image sequences using a robust motion estimation algorithm. In *Acoustics, Speech, and Signal Processing, 1991. ICASSP-91.*, 1991 International Conference on, pp. 2429-2432. IEEE Press. 1991.
- » [14] A. Tekalp, L. Murat, K. Mehmet, and M. Sezan. High-resolution image reconstruction from lower-resolution image sequences and space-varying image restoration. In *Acoustics, Speech, and Signal Processing, 1992. ICASSP-92.*, 1992 IEEE International Conference on, vol. 3, pp. 169-172. IEEE Press. 1992.
- » [15] OpenCV. Sitio opencv.org/, [/github.com/opencv/opencv](https://github.com/opencv/opencv), versión 4.8.0.76. 2017.
- » [16] Kivy. Sitio kivy.org, version: 2.2.1, 2023.
- » [17] Django. Sitio docs.djangoproject.com/en/4.2/, version 4.2, 2023.
- » [18] Flask. Sitio flask.palletsprojects.com/en/2.3.x, versión 2.3.3, 2023.
- » [19] Flet. Sitio flet.dev/docs, version 0.10.0, 2023.
- » [20] Y. Jia. Coffee. Sitio caffe.berkeleyvision.org/, version 1.0, 2014.
- » [21] J. Redmon. YOLO. Sitio pjreddie.com/darknet/yolo, version 3, 2018.
- » [22] Android studios. Sitio developer.android.com/studio, version Giraffe, Mar 1, 2023.
- » [23] Buildozer: Sitio buildozer.readthedocs.io/en/latest/installation.html, version 1.5.0, 2023.
- » [24] "Instalación del Buildozer y compilar un APK con Python y Kivy" del canal "Everardo MTZ", Jul 3.

2020.

- » [25] Labelme. Sitio github.com/wkentaro/labelme, versión 5.3.0, 2023.
- » [26] Caffe-SSD-Object-Detection de Amolik Vivian Paul. Sitio: github.com/amolikvivan/Caffe-SSD-Object-Detection. 2021.
- » [27] "Flet: build modern GUI for Opencv" del canal "Azu Technology", Jan 17, 2023.
- » [28] "Flet Tutorial - Build Android APK Release With Cordova" del canal "Sri Edy Nurcahyo", May 6, 2023.
- » [29] WSL, Sitio: learn.microsoft.com/es-es/windows/wsl/install, version 2, Aug 28, 2023.
- » [30] "Build & Deploy Kivy Python App (.apk) with Buildozer to Android (Windows & Linux)" del canal "Pygineer", Jan 31, 2022.
- » [31] Manjaro. Sitio manjaro.org, version 22.1.3, May 26, 2023.
- » [32] Arch Linux. Sitio archlinux.org, version 6.4.7, Aug 1, 2023.
- » [33] Ubuntu. Sitio ubuntu.com, version 22.04.2, Feb 23, 2023.
- » [34] I. Witten, E. Frank, M. Hall, C. Pal. Data Mining. Elsevier. 4th Edition. 2016.