

DETECCIÓN DE MÚLTIPLES OBJETOS EN UN ENTORNO PARA ASISTENCIA A INVIDENTES

Daniela López De Luise,^{1,2,3} Sung Park Jin¹, Silvia Hoferek¹, Nicolás Avila Lautaro³, Micaela Antonella Benitez³, Felix Raul Bordon Sbardella³, Rodrigo Iván Fantín³, Gastón Emmanuel Machado³, Aramis Oscar Mencia³, Anahí Ailén Ríos³, Rios Emiliano Luis³, Nahuel Edgardo Riveros³

daniela_idl@ieee.org

¹ CAETI – Universidad Abierta Interamericana – Facultad de Tecnología Informática

²CI2S Labs

³ Universidad de la Cuenca del Plata

Resumen

El objetivo de este artículo es presentar avances del prototipo para la asistencia a personas ciegas. El prototipo utiliza los dispositivos de un teléfono móvil estándar y su diseño minimiza el requerimiento computacional. El trabajo aquí presentado es el resultado del análisis en un entorno donde se presentan múltiples objetos y una serie de actividades que se realizaron para lograr el diseño que minimiza el requerimiento y mantiene un nivel de eficacia necesaria para el rendimiento del prototipo. No obstante, aún las actividades abarcan la recolección de videos, el pre-procesamiento para reducir el streaming, la segmentación de imágenes y el procesamiento con Minería de datos. El trabajo realizado según la necesidad planteada por personas invidentes videntes se propone un modelo específico de detección de objetos no presente en los modelos pre entrenados existentes y las diferencias obtenidas entre ambas. Asimismo, los modelos son expuestos a distintas técnicas de procesamiento para reducir el ruido e incrementar su eficacia en el reconocimiento del objeto. El dispositivo pretende asistir a las personas ciegas donde pueda analizar la circunstancia del entorno y poder comunicar de manera simple, concisa y con anticipación eventualidades que pongan en riesgo al usuario. El alcance del trabajo comprende los avances involucrados, nuevos hallazgos, características y procedimientos planteados para solventar peculiaridades detectadas y los resultados de los análisis de la misma.

Palabras clave Asistencia al invidente, Procesamiento de video, Detección de objetos, Minería de datos, Predicción de trayectorias.

Abstract

The aim of this article is to present advancements in the prototype for assisting blind individuals. The prototype utilizes standard mobile phone devices, and its design minimizes computational requirements. The work presented here is the result of analysis in an environment where multiple objects are encountered, and a series of activities were undertaken to achieve a design that minimizes requirements while maintaining the necessary level of effectiveness for the prototype's performance. Nonetheless, the activities still encompass video collection, pre-processing to reduce streaming, image segmentation, and processing with Data Mining. Addressing the need identified by visually impaired individuals, a specific object detection model not present in existing pretrained models is proposed, along with the differences observed between them. Additionally, the models are subjected to various processing techniques to reduce noise and increase their effectiveness in object recognition. The device aims to assist blind individuals by analyzing environmental circumstances and communicating potential risks to the user in a simple, concise manner, and with anticipation. The scope of the work includes the advancements made, new findings, features, and procedures proposed to address detected peculiarities, as well as the results of the analyses thereof.

Keywords Blind assistance, Video processing, Object detection, Data mining, Trajectory prediction

Introducción

Las personas que experimentan discapacidad visual a menudo se encuentran con obstáculos al intentar desplazarse en entornos pocos familiares, particularmente en espacios públicos y al aire libre. Para abordar esta problemática, y poder desplazarse en distintos lugares actualmente precisan el uso de herramienta como el bastón, el apoyo de otras personas o animales entrenados [01][02]. Actualmente el bastón es uno de los recursos limitados, pero más populares. Sin embargo, estas soluciones, salvo el caso de la ayuda de otras personas, no aseguran que el invidente esté a salvo ante algunos obstáculos como pozos o puertas.

Este trabajo está basado en el estudio comparativo de la optimización de performance en la detección instantánea con el apoyo de redes entrenadas adicionales, como herramienta de conceptualización que le permita al sistema HOLOTECH generar alarmas ubicuas para el invidente, ante situaciones donde las soluciones tradicionales no contemplan un mecanismo. La propuesta introducida en este proyecto, logra notificar al vidente mediante un mecanismo de toma de decisiones basado en sesgos de información evaluados automáticamente para detectar la presencia de potenciales riesgos específicos. La característica de esta herramienta es aprovechar tecnologías inteligentes y utilizar componentes accesibles al usuario. El dispositivo base es un teléfono celular gama media, eventualmente complementado con accesorios adicionales para mejorar la detección y predicción de eventos de interés [03][04][05]. La arquitectura del prototipo detecta obstáculos en el entorno cercano de la persona con discapacidad visual y genera patrones de sonido y vibración ante posibles riesgos. Está diseñado para ser montado en un dispositivo de uso frecuente por parte del usuario, su teléfono celular, facilitando el acceso a personas con recursos limitados al evitar la necesidad de realizar una inversión adicional.

El proyecto HOLOTECH tiene el propósito de respaldar y colaborar con individuos que presentan ciertas discapacidades visuales. En función de esto se estableció una colaboración

estrecha con el Circulo de No Videntes (CINOVI)[06], que permite establecer cuáles son los obstáculos de interés desde la perspectiva de los desafíos que enfrentan estas personas. La interacción con el usuario real ha permitido desmitificar hechos como la necesidad de detectar personas en el entorno, e identificar los objetos y eventos que sí constituyen los problemas más frecuentes del entorno, aquellos de detección más compleja mediante el bastón que persiste como el apoyo fundamental. Considerando los resultados anteriores del trabajo realizado en el proyecto con miembros de la Biblioteca Argentina para Ciegos (BAC)[07], del centro CAETI de UAI [08] y del equipo actual conjunto al CINOVI, se han definido objetivos que se enfocan en la detección de objetos utilizando sistemas inteligentes conexionistas y un sistema experto para interpretar los resultados obtenidos de las redes neuronales. De esta manera se obtienen comunicaciones ubicuas inteligentes que asisten al invidente con información clara y precisa.

El resto de este texto abarca el procedimiento de estudio de casos, análisis estadístico y conclusiones derivadas de la creación y evolución del prototipo. Es importante destacar que las progresivas habilidades incorporadas en el proyecto se centran en la problemática de identificar progresivamente diversos obstáculos quietos o móviles, que representan un riesgo potencial para las personas con discapacidad visual.

Etapa I. Elección de Metodología y Materiales

Para el desarrollo del prototipo se emplea software y hardware que cumple con las restricciones y requerimientos del análisis obtenido de las investigaciones realizadas por el intercambio con la BAC.

El conjunto de módulos de software que constituyen el proyecto detecta obstáculos y etapas que inician en la preparación esencial de los datos y culminan en el uso de modelos de comportamiento entrenados con sistemas conexionistas. Las etapas consisten en:

- A. Generación sistemática y protocolizada de datos a partir de material audiovisual. Esta primera etapa consiste en generar una base de datos

derivados de videos capturados en base a un protocolo predeterminado para acotar el sesgo de información a modelar con las redes neuronales. La duración promedio de las grabaciones es de 1 minuto. Para esta tarea se utilizan dispositivos móviles con capacidad de grabación de vídeo y memoria interna mínima de 64 GB. La grabación se realiza con el dispositivo a la altura del pecho, con un desplazamiento lento, y captando la mayor cantidad de posibles obstáculos que se encuentren durante el desplazamiento del sujeto. Estas capturas tienen una resolución estándar entre 320x240 dpi a 720x480 de dpi. El formato es XML [09].

- B. Pre-procesamiento: Una vez obtenidos los videos se tratan con el fin de mejorar la detección sistemática y consistente. Los procesos que se realizan durante esta etapa son: pasaje de imágenes a blanco y negro, manejo de blur y eliminación del contorno, filtrado de elementos [10][11] y por último reconstrucción [12].
- C. Entrenamiento del modelo neuronal con las imágenes: Como última etapa del sistema, se toman los datos ya procesados y clasificados, y se entrena a la red neuronal. Por medio de un proceso de optimización se ajustan los metadatos hasta obtener una tasa de error y métricas de validación aceptables.

En cuanto a las herramientas de software, se emplean el lenguaje Python por su capacidad de ser multiparadigma y adaptabilidad, OpenCV [13] para el entrenamiento conexionista, la librería Pandas de Python, Kivy como Framework y ambiente de trabajo [14], Flet [17] y Coffe [18] para el front-end de la aplicación, YOLO junto con el previamente mencionado OpenCV para el procesamiento de imágenes, Buildozer [19] como empaquetador de la APK [20], y Labelme [21] que se utiliza para crear nuevas etiquetas en el Log de rastreo.

Cabe mencionar que Androidstudios [22] es una la herramienta empleada inicialmente para probar ciertos apartados, pero por cuestiones

de compatibilidad, se descartó y fue cambiada por Buildozer.

Etapa II. Prototipo y Pruebas de laboratorio

La organización del ambiente de desarrollo del sistema dispone de carpetas que determinan qué videos son recortes y cuáles son escenas completas. A su vez se discriminan entornos cerrados de exteriores. Adicionalmente en el mismo grupo se identifican ambientes de acera, vereda y eventuales vehículos en el camino.

Para poder utilizar los videos, estos deben editarse a fin de eliminar las pistas de audio. De esta manera se evitan los datos que no son relevantes y se disminuye el peso del archivo. También se dividen algunos tests en segmentos menores, por caso, discriminando interiores de exteriores, si se cruza la calle, etc.

Gracias al CINOVI y los aportes en la información que compartió la entidad, se ha validado la lista de obstáculos y eventos de interés para el sistema en desarrollo. Debe destacarse que muchos de los elementos de interés son complicados de detectar ya que se encuentran por encima de la cintura.

Algunos de dichos elementos son:

- Autos mal estacionados en las veredas o en espacios no permitidos.
- 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.
- Rampas municipales.
- Dispositivos instalados, como acondicionadores de aire a alturas no recomendadas, principalmente cuando sobresalen de la pared.
- Pozos o roturas en el suelo, con o sin vallado.
- Obstáculos móviles eventualmente peligrosos como cochecitos, vehículos diversos, ciclistas, etc.

El Prototipo

El prototipo HOLOTECH está desarrollado con Kivy, "Caffe-SSD-Object-Detection" [23] y el framework Flet [24][25], para la detección de objetos y visualización móvil. Este trabajo presenta un avance en la generación de patrones optimizados para la identificación de los objetos escaneados. En cuanto a la creación del apk [26], se emplea Buildozer como en casos anteriores, que se basa en el sistema operativo Ubuntu[27] para su implementación.

Desarrollo de una prueba y armado de dato

Las pruebas emplean modelos predefinidos de OpenCv y entrenados con los datos específicamente colectados según el sesgo de interés. Es de destacar que estos modelos identifican objetos con base en las necesidades de las personas con visión normal. Surge la necesidad de detectar elementos del ámbito cotidiano, tales como puertas, que se hallan ausentes en los modelos predefinidos y por lo tanto se extienden los modelos existentes.

Este enfoque personalizado permite adaptar el sistema a las necesidades específicas de los usuarios con discapacidad visual, ampliando la gama de objetos detectables y mejorando la utilidad para el usuario en contexto de entornos cerrados y exteriores.

La construcción del nuevo modelo implica el uso de la herramienta *labelme* para etiquetar manualmente los objetos de interés, permitiendo así entrenar las características necesarias para la construcción del modelo y realizar inferencias en su detección de manera automática con un sesgo aceptable.

Al etiquetar manualmente los objetos de interés con *Labelme*, se logra capturar información detallada sobre las características distintivas de cada objeto con la máxima precisión. Esta información se convierte en datos de entrenamiento valioso, permitiendo que el modelo aprenda patrones visuales específicos asociados con dichos objetos.

La retroalimentación a lazo cerrado mediante el etiquetado y la incorporación de nuevas características a medida que se actualizan los conjuntos de datos permite mejorar la precisión

del modelo para realizar inferencias que permiten funcionar en escenarios diversos y desconocidos.

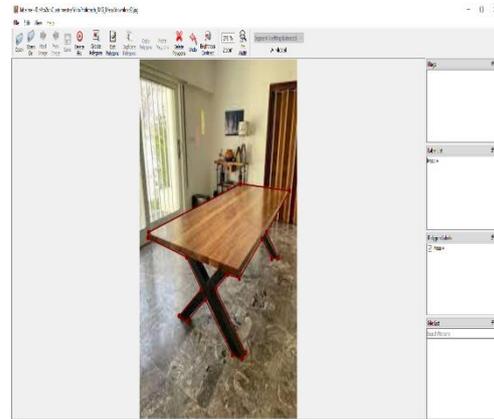


Figura 1 Interfaz gráfica del entorno Labelme

El modelo recibe los datos procesados y se entrena. Una vez creado se repite el procesamiento previo con otras imágenes siguiendo el mismo protocolo. Los resultados obtenidos con objetos introducidos a los nuevos modelos conexionistas se comparan con los otros resultados obtenidos previamente con OpenCV. La figura 2 provee la estructura de datos, diseñada para su análisis optimizando la detección desde las coordenadas relativas de ubicación y desplazamiento en el campo de captura del dispositivo. Fig. 3 Detección de sillas y encuadre en la región visual

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

Figura 2 Campos seleccionados para describir los objetos de estudio

Los datos obtenidos en estudios previos indican que es posible detectar objetos, desplazamientos y velocidades según sean las distintas tomas. También es posible observar la certeza de la detección de un mismo objeto. En la Fig. 2 se observan una silla con un marco de color lima y su correspondiente etiqueta ("chair").



Figura 3 Detección de sillas y encuadre en la región visual

En la Fig. 3 el patrón detecta una persona ("person"), pero como se ha indicado en secciones anteriores, los humanos detectados no son objeto de interés en nuestro contexto debido a que el invidente es capaz de detectarlos sin apoyo del sistema inteligente. Los movimientos, su ubicación y trayectoria sobre el espacio de captura del dispositivo también se deducen de manera automática a partir del desplazamiento del movimiento del propio usuario.

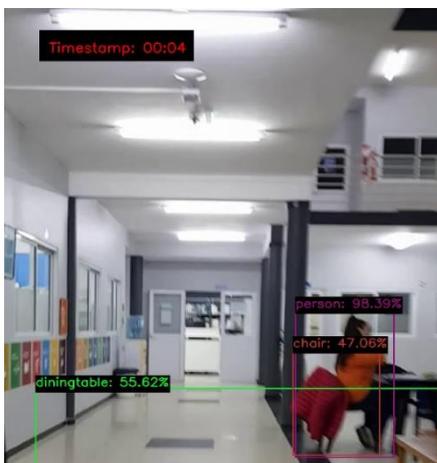


Figura 4 Detección de persona

La detección es el primer paso para que el sistema defina las zonas de riesgo, y ubicarlas espacio temporal en las circunstancias actuales del invidente a fin de evitar colisiones o accidentes. Como referencia relativa se define al propio usuario se re-calcula la distancia al evento de riesgo (Fig.4).

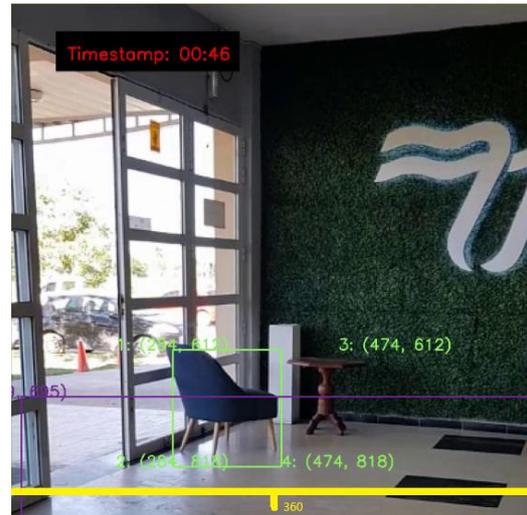


Figura 5 Coordenadas de una silla y sus marcaciones

Etapa III. Modelado del Sistema y Estadísticas Obtenidas

Para el entrenamiento del modelo se utilizaron 150 imágenes de sillas, puertas y mesas de las cuales todas se etiquetaron con Labelme. De este conjunto de imágenes, 120 fueron designadas al entrenamiento y 30 para validación. La Fig 6 y Fig 7 muestran objetos etiquetados y la predicción correspondiente con el modelo entrenado.

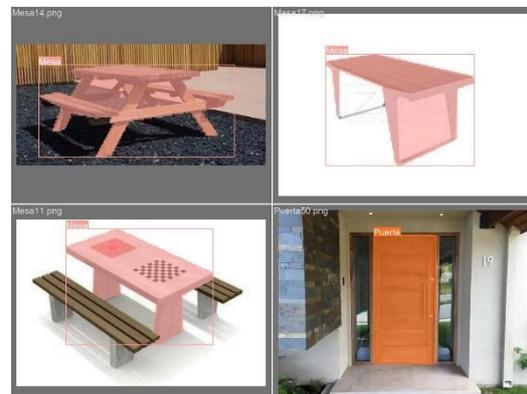


Figura 6 Datos etiquetados

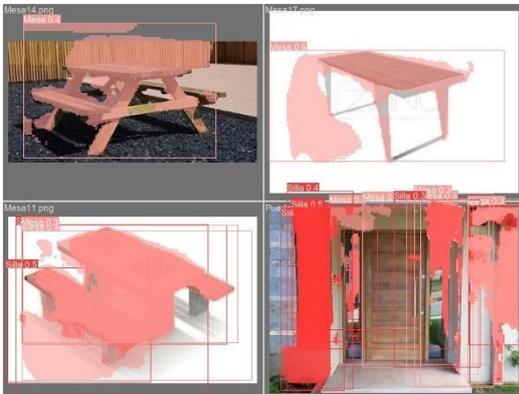


Figura 7 Predicciones con el modelo creado

Se puede apreciar que el modelo inicialmente creado es capaz de reconocer los objetos, aunque existen limitaciones en el sesgo con la presencia de objetos con similares características. Este tipo de limitaciones son las que se resuelven con la incorporación de nuevos entrenamientos.

No.	Label	Count	Weight
1	sofa	13	13.0
2	diningtable	296	296.0
3	bottle	15	15.0
4	chair	679	679.0
5	person	159	159.0
6	aeroplane	2	2.0
7	potteryplant	175	175.0
8	train	1	1.0
9	tvmonitor	7	7.0
10	boat	1	1.0

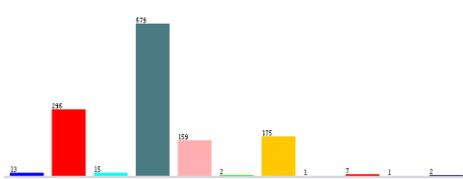


Figura 8 Capacidad de detección inicial

Es interesante observar la performance obtenida en versiones anteriores. La Fig. 8 muestra la eficiencia ante objetos genéricos básicamente con el criterio de las personas videntes para la mayoría de los casos salvo para las puertas.

Como es de esperar, el objeto de mayor interés para un vidente es de carácter diferente ya que no existe una necesidad de esquivar sino por el contrario, de determinar presencia de objetos. Los resultados ensayados determinaron 679 instancias de sillas, mesas (296), macetas (175) y personas (159) con precisiones centradas en los objetos de mayor frecuencia (Fig. 9).

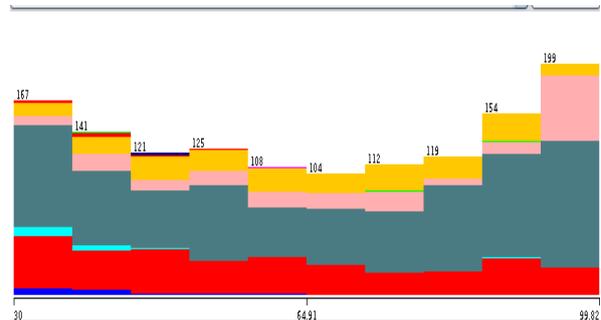


Figura 9 Precisión inicial para objetos

Como es de esperar, las clases mayoritarias tienen mayor precisión y estabilidad en el desempeño, pero están basadas en un entrenamiento de caras a las necesidades de los videntes. Esta característica ya fue reportada previamente usando el heurístico Density Based Clusterer de WEKA (DBC) [28], que clasifica en dos clusters distintos a las detecciones (Tablas 1 y 2 respectivamente para los de menor desempeño y los de mayor estabilidad) [29]

Cluster: 0 Prior probability: 0.2559	
Attribute	Normal Distribution. Mean (D)/Estimate
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 = 35)
match	(D) 60.4206 StdDev = 20.3299
timestamp	(E) 14 13 16 10 11 13 13 10 10 12 11 16 17 12 17 16 7 4 6 9 9 3 1 1 1 2 1 1 1 390)

Tabla 1 Características del Cluster de menor desempeño

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
object Name	(E) 1 297 6 1 13 1 31 2 2 1 1 (Total = 356)
match	(D) 67.6902 StdDev = 22.1793
timest amp	(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 24 31 31 31 31 (Total = 1050)

Tabla 2 Cluster de mayor estabilidad

Etapa IV. Conclusiones

Se concluye en base a los resultados obtenidos del modelo entrenado comparado con los modelos pre entrenados de OpenCV es necesario buscar alternativas que nos permitan disminuir la ambigüedad entre objetos muy similares, como entrenar y agregar objetos a filtrar que no pertenezcan al sesgo de interés y así lograr una detección más precisa.

No obstante, empleando los modelos pre entrenados para el desarrollo del prototipo se logró realizar un análisis de uso a partir de la detección de un obstáculo, además se logra definir detalles como la frecuencia con la se pueden obtener y analizar los diferentes datos obtenidos en tiempo real y la viabilidad de dispositivo móvil de gama media.

Bibliografía

- 1- 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.
- 2- 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.
- 3- J. S. Park, D. López De Luise, D. J. Hemanth & J. Pérez. Environment Description for Blind People. International Workshop Soft Computing Applications. Springer. 2018.
- 4- 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.

En conclusión, el prototipo ante la detección y seguimiento del obstáculo nos permite recolectar información en la cual no dan un entendimiento de la situación que el usuario puede estar ante una situación de riesgo. Sin embargo, como hay obstáculos no incluidos en los modelos pre entrenados entre los definidos por los videntes, se procede a la creación de nuevos modelos donde la eficacia en una primera etapa no es la esperada y es necesario un refinamiento del mismo.

Etapa V: Desarrollo a Futuro

A futuro se plantea poder realizar una mayor carga de objetos a la red neuronal, para poder mejorar y ampliar el espectro de detección que puede poseer el sistema. Para lograr esto, se está desarrollando la red neuronal por medio de las imágenes y la delimitación gracias a los nodos del labelme. Esto generó problemas a causa de que muchas de las imágenes poseían múltiples etiquetas y no solo el objeto importante. Por lo que se tiene pensado también a futuro cambiar todas estas etiquetas múltiples en imágenes para que posean sólo dos, una con el objeto y otra con la etiqueta "Background" para poder darle a entender al programa que es ruido y que no lo es. Además, una de las alternativas para reducir ambigüedades entre los objetos, se tiene planeado utilizar varios modelos y luego orquestar sus interacciones.

Se debe de evaluar también cuál de las opciones es la más eficiente para poder dar al usuario una señal de advertencia sobre los obstáculos, si es por medio de algún sonido o alarma, o es mejor por medio de ciertas vibraciones.

- 5- 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.
- 6-CINOVI. Asociación sin fines de lucro. Formosa, Argentina. Sitio www.cinovi.org.ar. 2023.
- 7- Biblioteca Argentina para Ciegos. Asociación sin fines de Lucro. Ciudad Autónoma de Buenos Aires, Argentina. Sitio bac.org.ar. 2023.
- 8- CAETI. Centro de Altos Estudios en Tecnologías Informáticas perteneciente a la Universidad Abierta Interamericana. Sitio caeti.uai.edu.ar. 2023.
- 9- Documentación de Python - 3.11.5 XML. Sitio:docs.python.org. 2023.
- 10- 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.
- 11- 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.
- 12- 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.
- 13- OpenCV. Sitio opencv.org/, [/github.com/opencv/opencv](https://github.com/opencv/opencv), versión 4.8.0.76. 2017.
- 14- Kivy. Sitio kivy.org, version: 2.2.1, 2023.
- 15- Django. Sitio docs.djangoproject.com/en/4.2/, version 4.2, 2023.
- 16- Flask. Sitio flask.palletsprojects.com/en/2.3.x, versión 2.3.3, 2023.
- 17- Flet. Sitio flet.dev/docs, version 0.10.0 , 2023.
- 18- Y. Jia. Coffee. Sitio caffe.berkeleyvision.org/, version 1.0, 2014.
- 19- Buildozer: Sitio buildozer.readthedocs.io/en/latest/installation.html, version 1.5.0, 2023.
- 20- "Instalación del Buildozer y compilar un APK con Python y Kivy" del canal "Everardo MTZ", Jul 3. 2020.
- 21- Labelme. Sitio github.com/wkentaro/labelme, versión 5.3.0, 2023.
- 22- Android studios. Sitio developer.android.com/studio, version Giraffe, Mar 1, 2023.
- 23- Caffe-SSD-Object-Detection de Amolik Vivian Paul. Sitio: [github.com/amolikvivian/ Caffe-SSD-Object-Detection](https://github.com/amolikvivian/Caffe-SSD-Object-Detection). 2021.
- 24- "Flet: build modern GUI for Opencv" del canal "Azu Technology", Jan 17, 2023.
- 25- "Flet Tutorial - Build Android APK Release With Cordova" del canal "Sri Edy Nurcahyo", May 6, 2023.
- 26- "Build & Deploy Kivy Python App (.apk) with Buildozer to Android (Windows & Linux)" del canal "Pygineer", Jan 31, 2022.
- 27- Ubuntu. Sitio ubuntu.com, versión 2.04.2, Feb 23, 2023.
- 28- I. Witten, E. Frank, M. Hall, C. Pal. Data Mining. Elsevier. 4th Edition. 2016.
- 29- Detección Automática de Objetos como asistencia a Personas Invidentes. Revista Abierta De Informática Aplicada. López De Luise, D., Park Jin , S., Hoferek , S., Avila Lautaro, N., Benitez Micaela, A., Bordon Sbardella, F. R., Fantín, R. I., Machado, G. E., Mencia Aramis, O., Ríos, A. A., Luis, E. L., & Riveros, N. E. (2023). Revista Abierta de Informática Aplicada (RAIA), 7(1), 37–50.