

**Complex & Intelligent Systems**  
**Development of an Electronic Travel Aid with Artificial Intelligence for Blind People in Public Areas**  
--Manuscript Draft--

<b>Manuscript Number:</b>	CAIS-D-23-00812	
<b>Full Title:</b>	Development of an Electronic Travel Aid with Artificial Intelligence for Blind People in Public Areas	
<b>Article Type:</b>	Original Article	
<b>Funding Information:</b>	Projects Dicyt 062117SG, Vicerrectoría de Investigación, Innovación y Creación, FONDEF No. ID21I10191 and STIC-AmSud AMSUD220026. (N.A)	Not applicable
<b>Abstract:</b>	<p>Under the global context, blindness and visual impairment (BVI) affects many individuals, limiting their mobility in public indoor environments due to crowds and a lack of accessible signage. The purpose of this paper is the development of an Electronic Travel Aid (ETA) that alerts users of obstacles and enables them to locate these signs in real-time using their hands and/or head. For this purpose, a dataset of signage was created, along with a portable prototype of the system incorporating stereoscopic vision, haptic/audible communication, and models for hand and sign detection. Improvements were achieved in the system's thermal behavior and its ability to generate simultaneous predictions at 9.31 fps using a distributed processing between the VPU and CPU of the system. It was effective in transmitting information to the user and robust against partial obstructions of up to 25% of the signage by the user's finger, maintaining a confidence range in the predictions from 66% to 72%, and reaching 97% in the absence of obstructions.</p>	
<b>Corresponding Author:</b>	<p>Nicolas Ibanez, Bachelor of Engineering Science  Universidad de Santiago de Chile  Santiago Centro, CHILE</p>	
<b>Corresponding Author Secondary Information:</b>		
<b>Corresponding Author's Institution:</b>	Universidad de Santiago de Chile	
<b>Corresponding Author's Secondary Institution:</b>		
<b>First Author:</b>	Nicolas Ibanez, Bachelor of Engineering Science	
<b>First Author Secondary Information:</b>		
<b>Order of Authors:</b>	<p>Nicolas Ibanez, Bachelor of Engineering Science  Ismael Soto, Ph.D.</p>	
<b>Order of Authors Secondary Information:</b>		
<b>Author Comments:</b>		
<b>Suggested Reviewers:</b>		

# Development of an Electronic Travel Aid with Artificial Intelligence for Blind People in Public Areas

Nicolás Ibáñez<sup>1</sup> and Ismael Soto<sup>1</sup>

<sup>1\*</sup> Departamento de Ingeniería Eléctrica, Centro de Investigación Multidisciplinario en Tecnologías de Telecomunicación (CIMTT), Universidad de Santiago de Chile, Av. Ecuador 3519, Santiago, 9170124, Región Metropolitana, Chile .

Contributing authors: [nicolas.ibanez.r@usach.cl](mailto:nicolas.ibanez.r@usach.cl);  
[ismael.soto@gmail.usach.cl](mailto:ismael.soto@gmail.usach.cl);

## Abstract

Under the global context, blindness and visual impairment (BVI) affects many individuals, limiting their mobility in public indoor environments due to crowds and a lack of accessible signage. The purpose of this paper is the development of an Electronic Travel Aid (ETA) that alerts users of obstacles and enables them to locate these signs in real-time using their hands. For this purpose, a dataset of signage was created, along with a portable prototype of the system incorporating stereoscopic vision, haptic/audible communication, and models for hand and sign detection. Improvements were achieved in the system's thermal behavior and its ability to generate simultaneous predictions at 9.31 fps using a distributed processing between the VPU and CPU of the system. It was effective in transmitting information to the user and robust against partial obstructions of up to 25% of the signage by the user's finger, maintaining a confidence range in the predictions from 66% to 72%, and reaching 97% in the absence of obstructions.

**Keywords:** Computer Vision; Deep neural networks; Stereo matching; Object detection; Mobility assistance; Blindness and visual impairments; Electronic Travel Aids; Accessibility; Ergonomic design

[Click here to view linked References](#)

1  
2  
3  
4  
5      Development of an Electronic Travel Aid with  
6      Artificial Intelligence for Blind People in Public  
7      Areas  
8  
9  
10  
11  
12  
13

### Abstract

14      Under the global context, blindness and visual impairment (BVI) affects many  
15      individuals, limiting their mobility in public indoor environments due to crowds  
16      and a lack of accessible signage. The purpose of this paper is the development  
17      of an Electronic Travel Aid (ETA) that alerts users of obstacles and enables  
18      them to locate these signs in real-time using their hands and/or head. For this  
19      purpose, a dataset of signage was created, along with a portable prototype of  
20      the system incorporating stereoscopic vision, haptic/audible communication, and  
21      models for hand and sign detection. Improvements were achieved in the system's  
22      thermal behavior and its ability to generate simultaneous predictions at 9.31  
23      fps using a distributed processing between the VPU and CPU of the system. It  
24      was effective in transmitting information to the user and robust against partial  
25      obstructions of up to 25% of the signage by the user's finger, maintaining a  
26      confidence range in the predictions from 66% to 72%, and reaching 97% in the  
27      absence of obstructions.

28      **Keywords:** Computer Vision; Deep neural networks; Stereo matching; Object  
29      detection; Mobility assistance; Blindness and visual impairments; Electronic Travel  
30      Aids; Accessibility; Ergonomic design  
31  
32  
33

## 34      1 Introduction 35

36      Blindness and Visual Impairment (**BVI**) poses a significant challenge in the daily lives  
37      of millions of individuals around the world, restricting their autonomy and mobility [1–  
38      3]. These limitations affect people of all ages, and their prevalence increases in parallel  
39      with the growth of the world population and the aging of the Chilean population.  
40      However, advances in technology are providing new opportunities for individuals with  
41      **BVI**, improving their quality of life and promoting their independence [4, 5].  
42

43      Within the context of technological solutions for individuals with **BVI**, various  
44      portable or wearable navigation systems have been developed over the past decades  
45      to assist those with **BVI** in orienting themselves in different environments and moving  
46      safely. The National Federation for the Blind (**NFB**) and the American Foundation  
47      for the Blind (**AFB**) have identified three main categories of aid systems for these  
48  
49

1 individuals: 1) vision enhancement, 2) vision replacement, and 3) vision substitution  
2 [6]. Within this last category, Electronic Travel Aids (**ETA**) stand out. These are elec-  
3 tronic devices that transform environmental information, which is usually transmitted  
4 through vision, into a form that can be conveyed through another sense, such as touch  
5 or hearing. The main function of an **ETA** system is to detect and locate objects along  
6 the user's path and provide as much information as possible, allowing them to deter-  
7 mine the range, direction, size, and height of objects. These devices are essential as  
8 they contribute to improving the mobility and safety of users with this condition.

9 In terms of need, individuals with **BVI** face significant difficulties when trying  
10 to navigate unfamiliar locations and move in public indoor spaces. These challenges  
11 become particularly pronounced in crowded places such as airports, metro stations,  
12 or shopping centers. Typically, in these indoor spaces, signage is designed according  
13 to the technical requirements of ADA Standards [7], which aim to guide visitors.  
14 However, most of these signs are positioned at a high level and do not feature a  
15 Braille system, hindering their accessibility to individuals with **BVI** from a distance.  
16 Because of these challenges, individuals with **BVI** often resort to the assistance of a  
17 guide, which is not always a viable solution as guides may not always be available  
18 or there might not be enough time to wait for their arrival. Hence, it is crucial to  
19 develop solutions that enable individuals with **BVI** to navigate independently in these  
20 spaces, avoiding moving obstacles and orienting themselves with real-time signage  
21 information. These solutions could leverage advancements in **AI** to better adapt to  
22 the specific needs of these individuals, optimizing their experience in unfamiliar and  
23 crowded environments.

24 In light of this need and advancements in **AI**, this article presents the devel-  
25 opment of an **AI**-based **ETA** system. Its purpose is to alert the user to potential  
26 obstacles and enable them to locate signage from a distance through the movement of  
27 their hand and/or head index, thereby facilitating their real-time orientation in these  
28 environments, as detailed in the following sections.

29 In section 2, a sociodemographic study of individuals with BVI was carried out  
30 to understand the challenges and needs they encounter daily, as well as the **AI**-based  
31 technological solutions proposed to address these challenges. This provides a solid  
32 foundation for the design and development of the proposed **ETA** system. Section 3  
33 details the methodology followed, which includes the selection and preparation of the  
34 indoor signage dataset, the training of object detection models, and the integration  
35 of the trained models into the **OAK-D** sensor's **VPU**. It also covers the configuration  
36 of the **OAK-D** sensor's DepthAI nodes and system measurements. Section 4 presents  
37 the design and construction of the proposed system, including the evaluation of the  
38 trained models, as well as the system's performance and thermal behavior. In section 5  
39 the main findings and contributions are discussed and finally the conclusions and  
40 the future work are presented in the sections 6 and 7.

## 43 2 Literature Review

44 This section 2 examines the current state of research and developments in the field of  
45 visual impairment, **ETA** systems, and relevant advances in **AI**. This field is increasingly  
46

relevant due to the growing global prevalence of **BVI**, with an emphasis on Latin America and especially in Chile, as discussed in subsection 2.1. The need to address this public health challenge with technological innovation leads to a comprehensive review of **ETA** systems and their integration with **AI**, described in subsection 2.2. This analysis sets the stage for a comparative evaluation, performed in subsection 2.3, that identifies current trends, opportunities, and challenges in the implementation of these systems.

## 2.1 Blindness and Visual Impairment

Blindness and visual impairment are physical conditions that affect the ability to see to various degrees, significantly impacting individuals' quality of life. According to a report conducted in June 2020 by the Association for Research in Vision and Ophthalmology (ARVO) [1], it was estimated with a 95% confidence interval (CI) that of the 7753 million inhabitants of the planet at that date, 49.1 million people were blind (54% women), 33.6 million had severe visual impairment (57% women), and 221.4 million lived with moderate visual impairment (55% women). The main causes of blindness and visual impairment, both severe and moderate, are uncorrected errors of cataracts, age-related macular degeneration, diabetic retinopathy, glaucoma, eye traction, myopia, hypermetropia, astigmatism, presbyopia, and amblyopia, with varying trends in prevalence and number of affected cases worldwide over time. Cataracts were the leading cause of blindness among people over 50 in 2020, followed by glaucoma, which increased between 1990 and 2010 [8], poorly corrected refractive errors, age-related macular degeneration, and diabetic retinopathy [2].

Visual impairment prevalence rapidly increases with advanced age, especially for women [3], and currently, the significant increase in the aging of the world population is expected, with a 95% CI, to result in the number of blind people increasing to 61.05 million worldwide by 2050 [9]. Therefore, despite advances in reducing the prevalence of blindness and visual impairment, it remains a significant global public health problem [10], resulting in an increase in the number of people affected by these causes of moderate or severe visual impairment [11, 12].

In Latin America, the prevalence of bilateral blindness varied between 1.3% in Buenos Aires in urban areas and 4.0% in two rural districts of Peru, with cataracts being the leading cause in older people, and retinopathy of prematurity the main cause of childhood blindness [13]. In Chile, according to the president of the Chilean Society of Ophthalmology (Sochciof), there are over 850,000 people with visual impairment, and of these, approximately 80,000 are blind, reaching 8.1% of the population in 2017, and it has been observed that 58.1% of the affected people are women compared to men, according to the "Informe de Desarrollo Social" 2022 [14]. Projections for 2050 estimate that the Chilean population will increase from 19.45 to 21.6 million inhabitants, of which 24% of the population will be people over 65 years of age, according to a World Bank study [15]. This corresponds to an increase of 11.9% since 2019 based on data from the National Institute of Statistics (INE) [16].

According to the latest results of the national disability study in Chile [17] conducted in 2015, this situation is closely related to the aging of people and their economic conditions, as the majority of this population has monthly incomes of CLP

1 377,000 on average, which are below the minimum wage, generally lower than the rest  
2 of the population, mainly due to their gradual retirement from the labor market [18].  
3 However, studies have also been conducted with children, where they examined 318  
4 of 421 children (76%) registered in 10 schools for the blind and found that 84% of the  
5 children examined in schools for the blind had severe visual loss, with retinopathy of  
6 prematurity responsible for 17.6% of these cases, and it was estimated that 49% of  
7 children with severe visual loss had preventable causes of blindness [19].

8 For these reasons, and taking into account the growing population of older adults  
9 in Chile, along with their economic and health challenges, as well as diseases like  
10 retinopathy of prematurity, the importance of addressing this issue to develop intel-  
11 ligent technologies is highlighted. These technologies, considering key factors such as  
12 accessibility, privacy, and cost, aim to make life easier for these individuals [20, 21],  
13 and face the economic barriers of many older adults in the country, to allow them to  
14 meet their basic needs.

## 15 2.2 Electronic Travel Aids and Artificial Intelligence

16 Over time, various technological solutions have been developed to improve the mobility  
17 and quality of life of blind and visually impaired people, exploring **ETA** systems,  
18 which have undergone significant evolution in recent decades with the advent of **AI**.

19 The first **ETA** systems focused on improving the functionality of the traditional  
20 white cane, incorporating mainly ultrasonic and infrared sensors, and microcontrollers  
21 into their design for detecting obstacles on the ground, resulting in fast, inexpensive,  
22 and easy-to-use systems [22, 23, 23–28]. However, due to limitations in obstacle detec-  
23 tion with proximity sensors and the lack of object recognition, wearable ETA systems  
24 were developed. Their designs incorporated haptic interfaces in hats or belts with  
25 vibrotactile actuators, GPS modules, and audible Text-To-Speech (**TTS**) communi-  
26 cation interfaces in the desired language with mobile devices such as Raspberry PI  
27 and smartphones, to improve information transmission to the user [29–32]. In terms  
28 of environment comprehension, **ETA** systems incorporated Computer Vision (**CV**)  
29 techniques such as Optical Character Recognition (**OCR**) for text recognition, the  
30 use of depth maps to estimate the distance to obstacles, image feature extraction like  
31 edges and corners with algorithms like SIFT, and recognition of people and objects  
32 of interest to users [33–40].

33 In recent years, object detection has advanced towards approaches based on Con-  
34 volutional Neural Networks (CNN) such as FAST R-CNN [41], MobileNetV2 [42],  
35 and the YOLO family, which have significantly improved computational efficiency  
36 and real-time performance compared to their predecessors. Notable is the YOLOv7  
37 paper [43], the current development of YOLOv8, and the improvements recently pro-  
38 posed, such as TPH-YOLOv5 [44] which improved YOLOv5's performance by 7%  
39 using an additional transformer-based prediction head to address specific challenges  
40 in object detection in scenarios captured by drones, and YOLOv7-DeepSORT [45]  
41 using YOLOv7 as the detection model and the DeepSORT algorithm, which improves  
42 the SORT (Simple Online and Realtime Tracking) algorithm [46], and contemplates  
43 tracking techniques, such as the Kalman Filter and the Hungarian algorithm [47],  
44

1 and a re-learning identities (ReID) model to extract appearance features of the multiple detected objects and compare them between frames improving the tracking performance.  
2

3 On the other hand, the use of approaches based on multiple specialized neural  
4 networks offers significant advantages in terms of accuracy, robustness, and efficiency  
5 in detecting body gestures. Models like OpenPose and MediaPipe Holistic provide  
6 integrated detection of the human body position in real time [48]. MediaPipe Holistic  
7 includes the BlazePalm Detector for hand detection and the Hand Landmark Model  
8 to identify key hand points [49], as well as Attention Mesh for face detection and  
9 3D facial mesh mapping [50]. These specialized approaches allow greater accuracy in  
10 identifying gestures and postures, which could prove useful in improving accessibility  
11 and interaction with electronic systems for people with disabilities.  
12

13 However, ETA systems face challenges and limitations in their implementation and  
14 real-time use. One of the main challenges is the processing of advanced CV algorithms  
15 in portable environments, which are computationally demanding, especially for mobile  
16 devices with limited processing and power resources [51, 52].  
17

18 Some designs of ETA systems have been proposed that circumvent the need for  
19 GPU-based hardware, avoiding its cost and the high energy consumption required by  
20 most deep learning algorithms for real-time inference, and instead employing state-of-  
21 the-art AI accelerators, in USB devices such as the Neural Compute Stick-2 (NCS2)  
22 or built into advanced depth sensors such as the OAK-D device. These technologies  
23 facilitate the use of model optimization techniques like OpenVINO and TensorFlow  
24 Lite, which allow the development of ETA systems with more specific approaches to  
25 improving the perception of the environment and the mobility of BVI individuals, such  
26 as the segmentation of sidewalks and streets to assess traffic conditions and abrupt  
27 elevation changes, and people detection to analyze crowds and contagion risks during  
28 a pandemic [53, 54].  
29

30 ETA systems have undergone continuous development in terms of local processing  
31 technologies, and new solutions based on mobile edge computing over 5G networks  
32 are being explored. A recent study [55] provides detailed and exhaustive simulation  
33 of edge computing for object detection with mmWave and sub-6-GHz connectivity in  
34 an application with high bandwidth and low latency requirements for an ETA system  
35 called VIS<sup>4</sup>ION. It consists of a backpack instrumented with multiple high-resolution  
36 cameras, local and edge AI processing, as well as haptic and auditory feedback. Despite  
37 the effort made to find the right balance between the quality and performance of the  
38 wireless connection and the processing capabilities, they could not meet the end-to-end  
39 delay requirements of 100 ms, which is critical to ensure real-time processing.  
40

41 On the other hand, some ETA systems have focused on improving user interaction  
42 with their environment through the use of gesture and posture detection techniques.  
43 The Point-to-Tell-and-Touch system [56] aims to give the user the ability to proprio-  
44 ceptively understand and locate the objects surrounding them. To achieve this, it uses  
45 an end-to-end neural network capable of predicting the location of the pointed object,  
46 its bounding box, and the object's category, as well as the location of the fingertip.  
47 Additionally, the system employs an Unsaturated Kalman Filter (UKF) to improve  
48  
49

1 performance and has hardware that consists of a Nvidia Jetson Box Tx2 microprocessor,  
2 a ZED Mini depth camera, an Arduino microcontroller, and headphones. Another  
3 similar but lighter and less costly system is Virtual Touch [57], which processes the  
4 image flow from the phone's camera, alternating between object detection and fin-  
5 gertip detection. It uses a convolutional neural network called TouchNet, designed to  
6 extract image features when the fingertip is not detected, and locate objects in nearby  
7 feature map areas with the fingertip when it is detected. Virtual Touch detects the  
8 fingertip using a skin color histogram approach to separate the hand from the back-  
9 ground, applies thresholding and filtering techniques to remove the background and  
10 enhance results, and identifies the hand contour and determines the convexity defect,  
11 the furthest point from the contour center, as the fingertip [58, 59].

12 Regarding obstacle detection and exploration of unknown environments, the HIDA  
13 and Corridor-Walker systems have shown promise in assisting BVI individuals. The  
14 HIDA system [60] is a portable solution designed to assist BVI individuals in explor-  
15 ing unknown environments. It consists of three hardware components: a lightweight  
16 LiDAR sensor, a laptop with GPU, and a bone conduction headset. The user per-  
17 forms an audio-guided environmental scan using the LiDAR sensor to capture detailed  
18 information about the environment in the form of point clouds. As the user moves  
19 and explores, the system determines their position on the point cloud map using the  
20 SLAM (Simultaneous Localization and Mapping) technique combined with odome-  
21 try. Subsequently, a 3D instance segmentation is performed on the captured point  
22 cloud, using a modified deep learning architecture based on PointGroup, which allows  
23 identifying and classifying individual objects present in the environment. Finally, a  
24 user interface processes the obtained semantic and instance information, providing  
25 acoustic feedback through the headset to help users avoid obstacles and locate spe-  
26 cific objects in their environment, thus enhancing their ability to safely and efficiently  
27 explore and navigate in unknown spaces. On the other hand, Corridor-Walker [61] is  
28 a smartphone-based system designed to assist blind individuals in navigating indoor  
29 hallways, avoiding obstacles and recognizing intersections. It uses a LiDAR sensor  
30 coupled to a smartphone to build a 2D occupancy grid map of the surrounding envi-  
31 ronment. Then, the system creates a path that avoids obstacles and detects upcoming  
32 intersections in the grid map. Finally, the system guides the user to follow the gen-  
33 erated path and notifies the user about the existence and shape of each intersection  
34 through vibration and audio feedback.

35 Table 1 presents a comparison of several innovative AI-based ETA systems, which  
36 incorporate CV algorithms and/or neural networks into their designs. The table high-  
37 lights essential aspects of each system, such as the hardware used for processing, the  
38 implementation of open-source software, and specific communication features for the  
39 user. Among these communication aspects, the use of obstacle alerts, voice synthe-  
40 sis, and a haptic interface in each system is detailed, with the aim of improving user  
41 interaction, comfort, and safety.

**Table 1:** Features of the most recent ETA systems based on AI

Ref.	Hardware Description	Open source software	Light design	Obstacle alerts	TTS	Haptic interface
[56]	NVIDIA Jetson TX2 NX micro-processor, ZED Mini stereo camera, headphones and an Arduino Due	x	x	x	✓	✓
[38]	Laptop, Arduino Mega 2560 microcontroller, Zed Camera, and a haptic belt	✓ YOLO9000	x	x	x	✓
[60]	Intel RealSense LiDAR L515 depth camera, laptop with GPU and headphones	x	x	✓	✓	✓
[57]	Headphones and a smartphone	✓ TouchNet	✓	x	✓	x
[53]	with a camera and GPU/CPU	✓ MobileNetV2	✓	x	x	✓
[54]	OAK-D device, AI accelerator Neural Compute Stick-2 (NCS2)	✓ SSD-MobileNet, ✓ DeepLabv3	x	x	x	✓
[55]	NVIDIA Jetson, 4 high-definition zed cameras, haptic belt and headphones	x	x	✓	x	✓
[61]	Smartphone with LiDAR sensor	x	✓	✓	✓	✓

### 2.3 Design Considerations for ETA Systems

Computer vision-based assistive solutions have proven promising for visually impaired individuals due to their affordability and accessibility [62]. Technologies such as intelligent computer vision sensors, with 3D perception cameras, can provide real-time environment information, enabling visually impaired users to make informed decisions about their surroundings at much greater distances than smart canes, and have gained popularity in recent years [63]. Implementing features such as distance to the next point of information and dynamic, real-time route instructions in navigation assistance systems has been effective in improving user confidence and reducing user anxiety [64, 65]. On the other hand, the incorporation of multimodal interaction, involving auditory and tactile modalities, in portable and accessible navigation systems for people with BVI can significantly enhance their navigation experience and provide a comfortable and efficient usage environment [66].

When designing assistive devices, it is crucial to consider the user-system interaction, including aspects such as hands-free, ears-free, wearability, and ease of use [6]. Additionally, it is also important to take into account in their design the ergonomic temperature limits to ensure user comfort; for example, the limits for plastic and aluminum housings are approximately 45°C and 41°C, respectively, due to the differences in heat transfer [67].

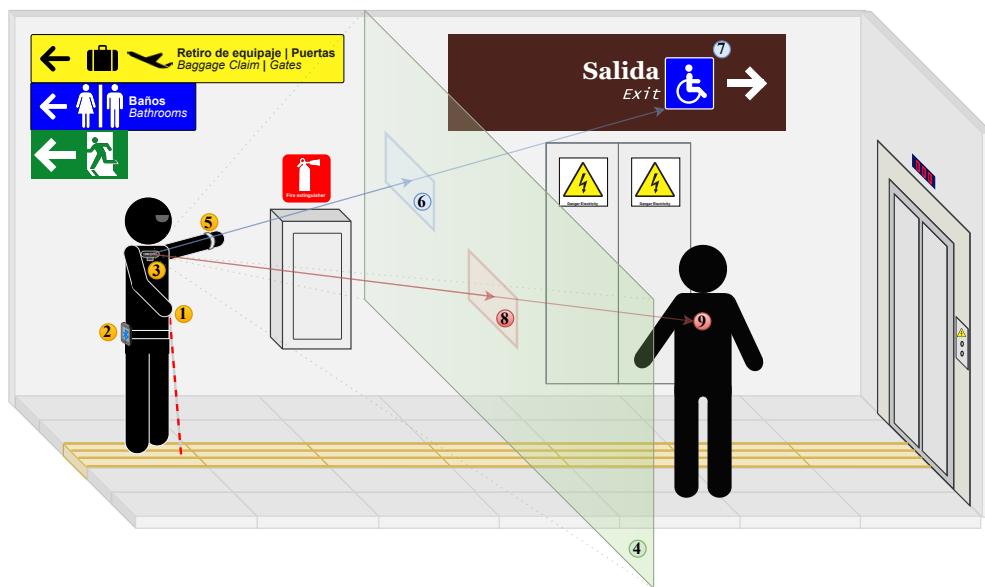
### 3 Methodology

In this section, the methodology adopted for the development of a functional prototype of an **ETA** system based on **AI** intended for people with **BVI** is presented. Technical and theoretical aspects of the tools and techniques implemented, selected for their relevance and effectiveness in this field of study, will be covered. Section 3.1 will provide a detailed description of the proposed system to be developed in this article, followed

by section 3.2 which will present the tools and techniques used for the development of the prototype system with the training, implementation, and evaluation of the **CV** and **DL** algorithms that support its operation. A discussion on the evaluation methods and metrics employed to measure the performance of the system is included. This methodological approach has been chosen with the aim of ensuring the accuracy, robustness, and efficiency of the proposed system.

### 3.1 Description of the proposed system

The development of this system focused on improving the cognitive orientation of people with **BVI** and assisting them in finding facilities such as bathrooms, restaurants, and preferred seating on public transportation, or anything that is signposted. The idea of the proposed system present in Figure 1, is that it can be used in situations such as, for example, at an airport where people around do not speak their language and cannot tell them where the boarding gates, baggage claim or emergency exits are, which are vital in emergency situations such as fires or earthquakes.



**Fig. 1:** Illustration of the use of the proposed system

In Figure 1, a generic situation is illustrated in which a person with **BVI** (1) can be seen using the proposed **ETA** system, composed of a smart sensor (3) that can perform hand detections and indoor signage detections at the same time along with depth estimates of the sensor's field of view (4). This information is sent to a microprocessor or mobile device (2) that is responsible for finding the closest detection of signage (7) to a virtual reference (6) whose position can be controlled with the detections of a user's hand (5), or it can remain fixed in the center. On the other

hand, the microprocessor is responsible for calculating the distance to the obstacles (9) that are within an area of interest (8). If these obstacles are close enough to be dangerous, the microprocessor sends control signals to a user communication interface to emit short, more frequent sounds as the user approaches them, similarly, vibrations are made when the virtual reference approaches the nearest signage in the cardinal directions up, down, left or right. Finally, when the virtual reference is in the vicinity of the nearest signage, the microprocessor generates a contextual **TTS** phrase with the information of the nearest signage and reproduces it through a headset.

### 3.2 Synthesized tools and techniques used

In the proposal of this system, the **AI** processing device **OAK-D** is incorporated, capable of providing real-time depth data and running **CV** algorithms. **YOLO** algorithms are applied to detect signage in the user's environment, complemented with Mediapipe's Palm Detection and Hand Landmark models for tracking the user's hands. For training the detection models, Google Colab and Roboflow tools are used, allowing proper preparation of data sets and accelerated training of models. Finally, the effectiveness and robustness of the system are assessed using the following specific metrics and performance tests.

### 3.3 Methods for evaluating the performance of the system

In this section, the evaluation methods and metrics used to measure the performance of the system developed in this article in two independent experiments described below are discussed.

#### 3.3.1 Performance evaluation of object detection models

The following are the different metrics used in this article to evaluate the performance of object detection models, both generally and specifically on the **OAK-D** device.

#### 3.3.2 Average frame rate

This experiment is carried out with the aim of determining the performance of the object detection models on the **OAK-D** device, and determining which is the appropriate configuration to be used in the system developed in this article.

As a prerequisite, the object detection models must be converted to the OpenVINO format, for which the Luxonis **YOLO** model conversion platform is used, the link to which can be found in Appendix D.3. Once the models are converted, the experiment to evaluate the performance of the object detection models on the **OAK-D** device is carried out.

To evaluate the performance of the object detection models on the **OAK-D** device, an experiment is performed in which the average frames per second (FPS) are counted in 1000 iterations. During each iteration, the time it takes for the model to predict a class and for that prediction data to arrive at the host device is obtained. At the same time, successful detections, failed detections, and undetected detections are counted with a test object placed in front of the **OAK-D** device.

1           The metrics used in this experiment are:  
2           • **Average FPS**: Represents the average number of frames processed per second.  
3           • **Successful detections (TP)**: Number of correct detections of the test object.  
4           • **Failed detections (FP)**: Number of incorrect detections of the test object.  
5           • **Undetected detections (FN)**: Number of times detections were not made due to  
6           low prediction confidence.

### 8           3.3.3 Confusion Matrix 9

10          A confusion matrix is a graphical or tabular representation of a classifier model, and  
11          it is named as such because it allows us to see if the model is confusing classes with  
12          each other. The confusion matrix is generated by comparing the labels or annotations  
13          of the test data set with the predictions made by the model trained with the training  
14          and validation data sets.

15          In the general case of an object detection model with  $N$  different classes, the con-  
16          fusion matrix has a dimension of  $N \times N$ , where each class has its own row and column,  
17          which represent the predictions made by the model and the labels of the test data set,  
18          respectively. The values on the main diagonal represent the correct predictions for  
19          each class (true positives **TP**), while the values off the main diagonal represent incor-  
20          rect predictions (false positives **FP** and false negatives **FN**). In addition, the confusion  
21          matrix includes at the bottom and right side, a row and a column of length  $N$  for  
22          the **Background NF** and **Background FP** classes, respectively. A **Background FP** (false  
23          positive background) refers to background objects of an image that do not belong to  
24          any of the classes, but are predicted as one of them. Conversely, a **Background FN**  
25          (false negative background) refers to classes that are not image background objects,  
26          but are predicted as background objects.

### 28          3.3.4 Mean Average Precision (mAP) 29

30          Throughout this article, the **mAP@.5** metric (Mean Average Precision at 50% Intersec-  
31          tion over Union) is used, popularized in the context of object detection by competitions  
32          and datasets like PASCAL VOC [41] and COCO [68]. This evaluation metric is widely  
33          adopted by the **CV** research community to evaluate the solidity of object detection  
34          models, such as Fast **R-CNN**, **YOLO** and Mask **R-CNN**. The **mAP@.5** metric eval-  
35          uates the accuracy of a model in terms of class prediction and localization of their  
36          respective bounding boxes, using similar concepts to the confusion matrix, such as  
37          true positives (**TP**), false positives (**FP**) and false negatives (**FN**) to calculate pre-  
38          cision and recall at different confidence levels. The metric employs Intersection over  
39          Union (**IoU**) to determine whether a detection is considered a **TP** or **FP**, based on a  
40          50% **IoU** threshold in the case of **mAP@.5**.

41          In this work, the value of **mAP@.5** is directly extracted from the training results  
42          of the **YOLO** object detection models. A detailed description of the calculation of this  
43          **mAP** metric can be found in the article [69].

### 3.3.5 Evaluation of the Thermal Behavior of the System

Considering that the ergonomic temperature limits for portable devices are 41°C for aluminum surfaces [67] and that the maximum operating temperature of the **VPU** of the **OAK-D** device is 105°C, in this article two independent experiments are carried out to evaluate and compare the thermal behavior of the system based on two different approaches for processing neural networks, described below:

- **System without distributed processing of neural networks:** The system is configured in such a way that all predictions of the neural networks are processed in the **VPU** of the **OAK-D** device.
- **System with distributed processing of neural networks:** The system is configured in such a way that some predictions of the neural networks are processed in the **VPU** of the **OAK-D** device and others in the **CPU** of the host device.

The aim of both experiments is to identify a suitable configuration for neural network processing, determine if the smart sensor is suitable for use near the user's skin and finally, evaluate if it is necessary to implement an additional cooling system for the **OAK-D** device.

The system's temperature measurements and their requirements are detailed below:

- **Measurement of the temperature of the VPU of the OAK-D device:** To measure the temperature of the **VPU** of the **OAK-D** device, it is necessary to previously configure the SystemLogger node using the DepthAI API, the documentation of which is available in the supplementary material [D.2](#).
- **Measurement of the temperature of the heatsink of the OAK-D device:** To measure the temperature of the heatsink of the **OAK-D** device, a digital thermocouple is used, fixed to the heatsink of the **OAK-D** device, whose measured data are transmitted to the host device by wire through the SPI serial communication protocol. In addition, it is necessary to use a custom library called MAX6675, which is available in the repository of this article.
- **Measurement of the temperature of the CPU of the host device:** The temperature of the **CPU** of the host device can be obtained directly using Python's subprocess library, taking advantage that most microprocessors measure the temperature of the **CPU** as a subprocess of the operating system.

These measurements allow evaluating the thermal behavior of the system based on different approaches to incorporate neural networks in the relevant processing units.

### 3.3.6 Evaluation of the Robustness and Communication of the System

The aim of this methodology is to evaluate the performance of the system under real operating conditions, analyzing both the detection of signage under user disturbances and the effectiveness of the communication interface in providing haptic and audible feedback. To achieve this, the following methodology is proposed:

#### 1. System tests for user hand disturbances

The aim of this test is to evaluate how the system maintains the detection of signage when the user interferes with their own hand, allowing the analysis of the system's resilience and adaptability in obstruction situations.

- **System configuration:** Configure the system with the best object detection model obtained in section 3.3.1, and with the best form of the neural network procedure described in section 4.4.
- **Recording the frame rate:** Record the average frame rate during the detection of signage and the user's finger, to analyze the real-time performance of the system.
- **Detection confidence recording:** Partially disturb different test signage with the user's finger, in a 25% and 50% obstruction, and record the detection confidence in each case, with the aim to analyze how the system adapts to user interferences.

## 2. Communication interface tests

The aim of this test is to evaluate the behavior of the communication interface and its ability to provide effective haptic and audible feedback, allowing the analysis of the quality of the information provided to the user and the ability of the system to guide the user appropriately.

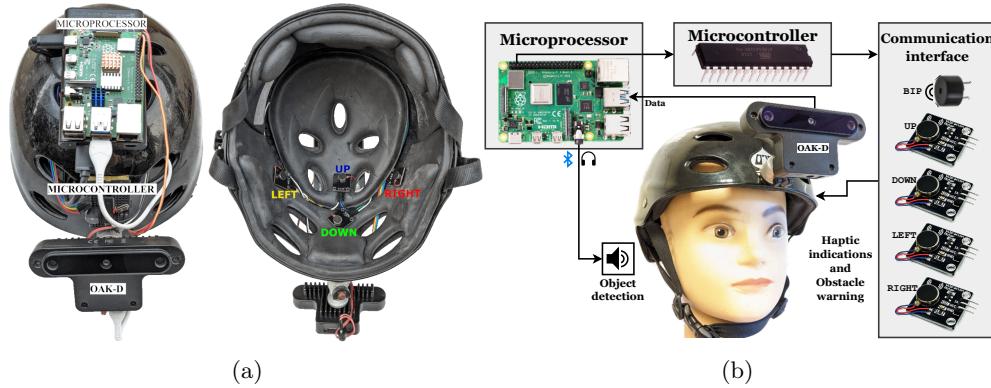
- **System configuration:** Configure the system with the best object detection model obtained in section 3.3.1, but without processing the hand detections, to keep the virtual reference fixed in the center of the frame, and not affect the tracking of the nearest detection.
- **Recording the frame rate:** Record the average frame rate during the detection of signage and the user's finger, to analyze the real-time performance of the system.
- **Recording of the virtual reference and test detection:** Record the position of the virtual reference and test detection in each frame, with the aim to analyze the accuracy and stability of the detections.
- **Proximity alert recording:** Record the proximity alert generated by the communication interface in each frame, in order to analyze how the system provides feedback to the user based on the proximity to obstacles and the effectiveness of the haptic and audible indications.

## 4 Results

This section presents the results obtained during the development of this work. Section 3.1 describes the implementation of the hardware and software of the proposed system, including the prototype design, the electrical circuit, and the flowchart of the main program. Later, in section 4.2, the details of the dataset created for the training of object detection models are presented, along with the results of the evaluations carried out to measure the accuracy of these models. In section 4.3, the evaluations conducted to measure the processing speed of the system, its robustness against disturbances, and its thermal behavior under different configurations are described. Finally, section 5 discusses the results obtained.

## 4.1 Implementation of a Prototype for the Proposed System

To carry out the implementation of the proposed system described in subsection 3.1, a system diagram was designed, illustrated in Fig. 2b, which shows the used hardware components and the data flow between the smart sensor, the microprocessor, and the user communication interface. Based on this diagram, a test prototype was created, which is presented in Fig. 2a.



**Fig. 2:** Hardware implementation of the system: (a) Test prototype, (b) System diagram

### 4.1.1 Hardware Components of the Prototype

A brief description of the components and their use in the test prototype, as shown in Fig. 2a, is presented below:

- **OAK-D Smart Sensor:** This sensor is responsible for capturing images and performing real-time detection analysis.
- **Raspberry Pi 4 Model B (Host):** High-performance microprocessor with a quad-core 64-bit ARM-Cortex A72 at 1.5 GHz and 4 GB of LPDDR4 RAM memory. It processes the information from the smart sensor and communicates with the user communication interface and other system components.
- **Microcontroller (Atmega328p):** Responsible for independently controlling the vibrators and the buzzer, receiving instructions from the microprocessor through the UART communication protocol.
- **PWM Switch DC Vibrating Motor (Vibrators):** A set of four 3.3V PWM vibrating motors is used to form the haptic interface of the system. They generate directional vibrations in cardinal directions to inform the user about the location of the signage in their environment.
- **Headphones:** Wireless electroacoustic transducers connected via Bluetooth to the microprocessor that inform the user about the information obtained by the system in real time.

- 1  
2  
3  
4  
5 • **Power supply:** 5V/2A lithium-ion portable battery to supply power to the entire  
6 system efficiently and reliably.  
7  
8

9 **4.1.2 Description of the Prototype Design**  
10  
11  
12  
13  
14

15 The prototype design, illustrated in Fig. 2a, consists of an ergonomic and adaptable  
16 sports helmet, equipped with an adjustable support to place the OAK-D sensor at  
17 the front. In this way, it ensures that the sensor is aligned with the user's field of  
18 vision and can capture images properly. This smart sensor connects to a Raspberry  
19 Pi 4 Model B microprocessor via a USB-C cable for communication and power.  
20

21 The Raspberry Pi 4 Model B microprocessor and the 5V/2A lithium-ion portable  
22 battery are mounted compactly on top of each other on the front of the sports helmet.  
23 This allows easy access to the components and a balanced weight distribution. The  
24 Atmega328p microcontroller is located on a protoboard, where the vibrators, buzzer,  
25 and the necessary cables for communication and power supply are connected.

26 Regarding the haptic design of the user communication interface, four vibrators  
27 were placed inside the helmet in the shape of a cross, following the cardinal directions  
28 (up, down, left, and right) as illustrated on the right of Fig. 2a. In this way, the  
29 user can perceive vibrations that indicate the location of the signage detected by the  
30 system, which facilitates orientation and navigation in the environment.  
31

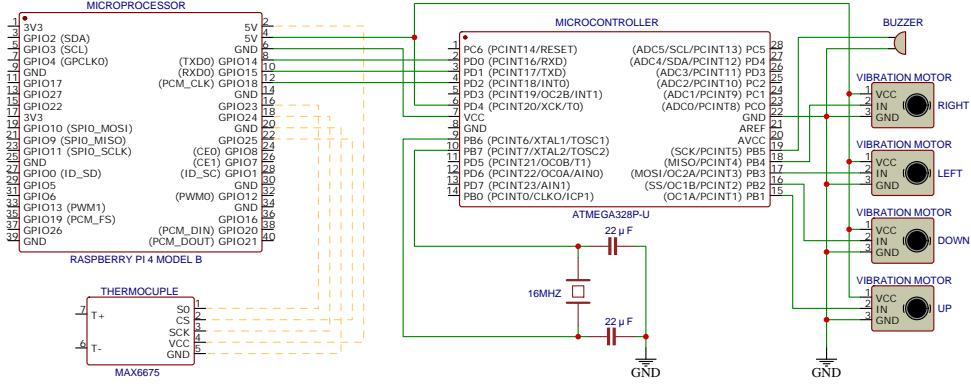
32 For the auditory interface, two wireless headphones were integrated, connected to  
33 the microprocessor via Bluetooth, which provide detailed and real-time information  
34 about the data obtained by the system. Also, a buzzer was incorporated to emit  
35 warning sounds in the presence of obstacles, increasing the user's safety during their  
36 displacement.  
37

38 **4.1.3 Description of the Prototype's Electrical Circuit**  
39  
40

41 The electrical circuit of the test prototype is presented in Fig. 3, where the connection  
42 between the different system components and their distribution in the helmet can  
43 be observed. The implementation of this circuit ensures efficient and reliable communication  
44 between the components, as well as adequate power supply for the correct  
45 operation of the system.  
46

47 The circuit illustrated in Fig. 3 features several key components and connections  
48 that facilitate the operation and communication between the different elements of the  
49 system. Some of these components and connections are explained in detail below:  
50

- 51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65
- **Oscillator Circuit:** The 16MHz quartz crystal is a resonator that, together with the 22pF capacitors, forms an oscillator circuit. This circuit is responsible for generating and stabilizing the ATMEGA328P microcontroller's clock signal at a constant frequency and phase of 16 million cycles per second. The combination of these components ensures stable and precise operation of the microcontroller.
  - **MAX6675 Thermocouple:** Although the thermocouple is not part of the system itself, it was connected to the Raspberry Pi 4 microprocessor to evaluate the system's thermal behavior, as described in subsection 3.3.5 of the methodology. A short-distance wired connection was made using the synchronous SPI (Serial Peripheral Interface) communication protocol. The MAX6675 thermocouple acts



**Fig. 3:** Electrical circuit of the test prototype

as a slave device in this communication, while the Raspberry Pi 4 is the master device that controls the communication through program [B.3](#) to extract the digital information of the temperature measured in the calibrated thermocouple.

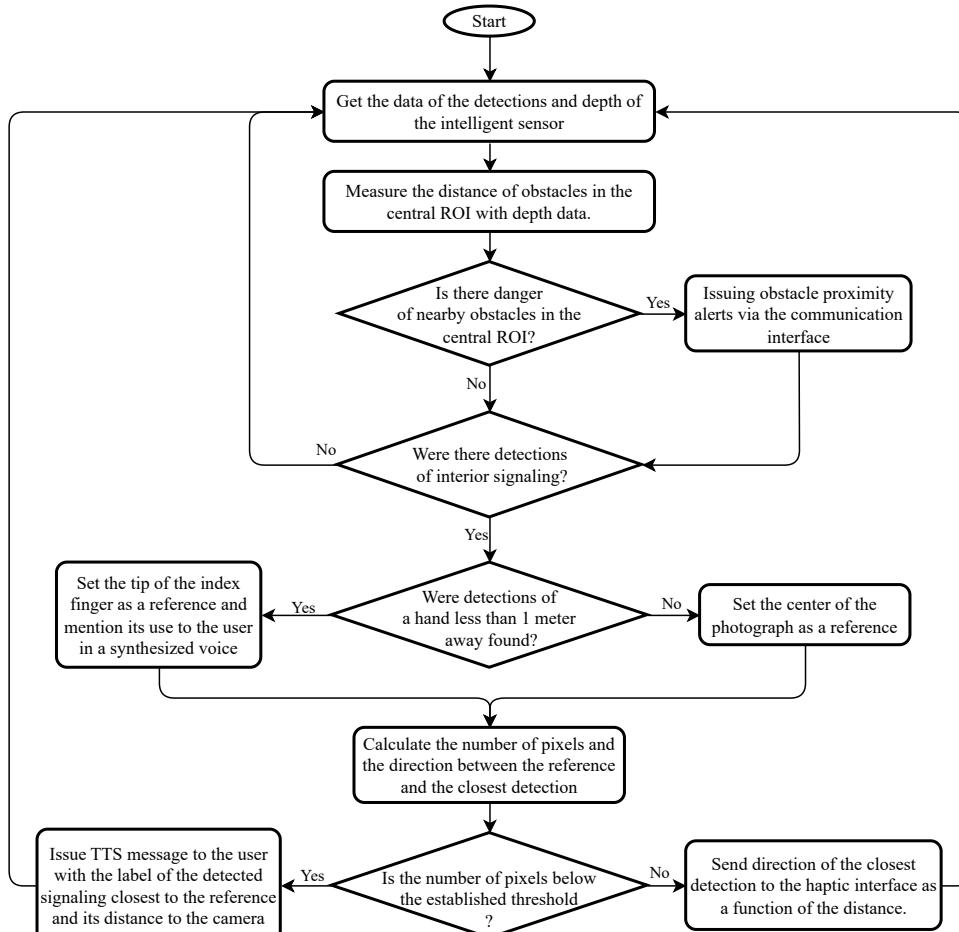
- **Communication between the microcontroller and the microprocessor:** Communication between the Atmega328p microcontroller and the Raspberry Pi 4 microprocessor was established using the [UART](#) (Universal Asynchronous Receiver-Transmitter) serial communication protocol. To achieve this communication, only the Rx receiver pin of the Atmega328p was connected to the Tx transmitter pin of the Raspberry Pi 4, since the microcontroller only receives a byte of information from the microprocessor to control the vibrators and the buzzer. The communication was carried out using ASCII characters through the microcontroller's main program [B.4.1](#) and the program compiled in the microcontroller [B.4.2](#). This connection allowed efficient information exchange between both devices and facilitated the control of system components.

These components and connections form the foundation of the test prototype's electrical circuit and allow the efficient and accurate operation of the system as a whole. By understanding how these elements work and how they connect with each other, a better insight into the prototype's inner workings is gained, and adjustments or improvements can be made as needed.

#### 4.1.4 Description of the Prototype's Software

The flowchart shown in Fig. 4 illustrates the iterative process carried out by the microcontroller when communicating with the OAK-D smart sensor and the communication interface microcontroller. This process allows TTS messages to be sent to the user with contextual information on the nearest signage to the virtual reference, to transmit cardinal haptic signals based on the number of pixels between the virtual reference and the center of the nearest detected signage, and to generate proximity alerts based on the distance to obstacles measured in the central region of interest (ROI). To establish the virtual reference, the distance between the user's arm and

the **OAK-D** sensor was considered, ensuring that the detected hand is the user's and conditioning how this reference is defined. This diagram represents the operation of the main program, which is detailed in Appendix [B.4.2](#).



**Fig. 4:** Flowchart

#### 4.1.5 Configuration of DepthAI Nodes in the OAK-D Device

Initially, the Host device, in this case, a Raspberry Pi 4, starts the main program [B.4.2](#), which aims to establish communication with the **OAK-D** device through the DepthAI API. This API allows the configuration of DepthAI nodes, as described in its documentation [D.2](#). To carry out this task, the **DepthYoloHandTracker** class is used, specifically designed to facilitate the management and configuration of DepthAI

nodes, and which can be found in Appendix B.1. Below is a complementary description of the DepthAI nodes configuration used in the test prototype:

- **ColorCamera and MonoCamera Nodes:** The ColorCamera node was used to capture frames with a resolution of 640x640 pixels in BGR format and the MonoCamera node to capture two monochromatic images with a resolution of 400x400 pixels with different perspectives for generating a depth map with the StereoDepth node. These nodes were configured to capture images at a frequency of 30Hz.
- **StereoDepth Node:** The StereoDepth node was configured to generate depth maps using the stereoscopic matching algorithm, with a high-density profile (HIGH\_DENSITY). Also, the output of the depth map was adjusted to be aligned with the RGB camera's perspective to facilitate the depth estimation of the YOLO neural network's color detections. Finally, a depth range of 0.5 to 10 meters was established to focus and improve the depth estimation accuracy.
- **YoloDetectionNetwork Node:** This node was used to run the YOLO neural network models on the OAK-D device. The models were previously converted into OpenVINO format using the platform in Appendix D.3. Later, the specific parameters of the trained models were configured, such as the path of the models with the new .blob extension, the total number of classes, and the dimensions of the input image. In addition, confidence and intersection over union (IoU) thresholds were set to 0.6 and 0.5, respectively, to filter out detections with low confidence.
- **NeuralNetwork and ImageManip Nodes:** The NeuralNetwork node was used to integrate the Palm Detection and Hand Landmarks neural models, which require input images with resolutions of 128x128 and 256x256 pixels, respectively. To meet these requirements, the ImageManip node was configured to resize the input images according to the dimensions required for each model. The NeuralNetwork and ImageManip nodes are created only when the use\_hand method is set to True in the DepthYoloHandTracker class.
- **SystemLogger Node:** The SystemLogger node was used to monitor the temperature of the OAK-D device's VPU. This node was necessary to carry out the test to evaluate the system's thermal behavior described in subsection 3.3.5 of the methodology. This node is created only when the temperature\_sensing method is set to True in the DepthYoloHandTracker class.
- **XLinkIn, XLinkOut, and XLink Nodes:** These nodes were used to establish communication between the host device (Raspberry Pi 4) and the OAK-D device. Several XLinkOut outputs were configured to transmit the RGB images, the neural network detections, the mapping of bounding boxes, and the depth map to the host device.

Fig. 5 presents the connection diagram of the DepthAI nodes configured in the OAK-D device, illustrating how they are interconnected and work together to carry out the required image processing and analysis tasks in the prototype.

## 4.2 Training of object detection models

In order to use the fastest and most accurate object detection model on the OAK-D device for detecting indoor public signage, four YOLO models were trained with

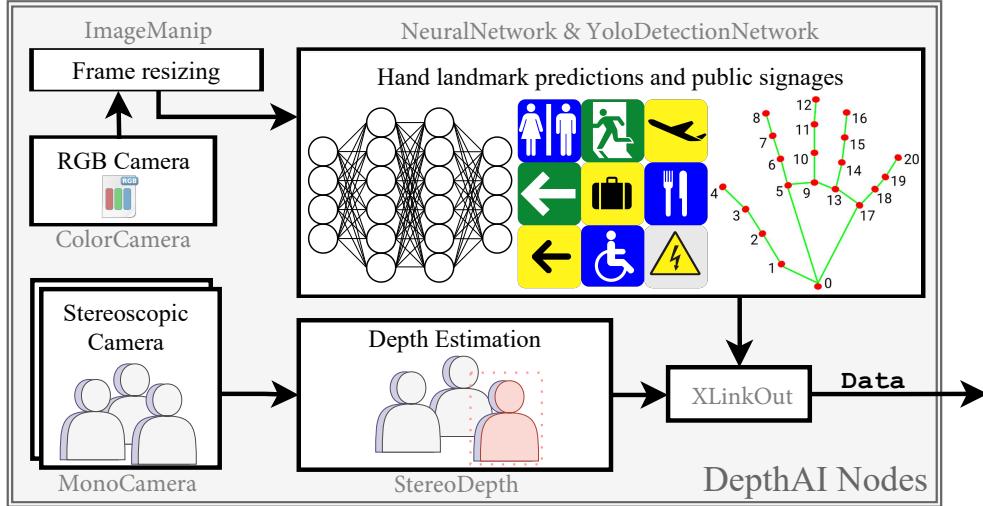


Fig. 5: Configuration diagram of DepthAI nodes of the OAK-D device

the same 640x640 pixel input, but with different neural network architectures. The trained models were YOLOv5n, YOLOv7t, YOLOv7s, and YOLOv8n, which is still in the development stage. The models were trained with the indoor signage dataset, and exported in OpenVINO format for use on the [OAK-D](#) device, which are available in this work's GitHub repository described in Appendix A [70]. Below, the details of the procedure carried out from data acquisition to the training and evaluation of the models on the [OAK-D](#) device are described in the following subsections.

#### 4.2.1 Creation of the dataset for public indoor signage

For data acquisition, a 12-megapixel camera was used to capture photographs and videos of signage in different airports, metro stations, and supermarkets, with various perspectives and lighting conditions. Frames were selected from each video to represent a wide range of situations and ensure a diverse and representative dataset. Once the images were collected, a manual labeling process was carried out using object annotation tools from the web platform Roboflow, available in Appendix D.5. Pre-processing techniques were used, such as resizing and normalizing images, to ensure compatibility and optimal performance in the training of the 640x640 pixel [YOLO](#) model. Roboflow's data augmentation function was also used to generate additional images with variations in rotation, brightness, and contrast changes to increase accuracy and reduce overfitting in trained models. Signage images were not flipped or horizontally rotated, as this could confuse the model when detecting them, especially with directional signage. Finally, a total of 8620 object annotations were obtained in 6807 images, which were grouped into 13 classes of typical signage from different airports and metro stations. It's important to mention that the dataset resulted in

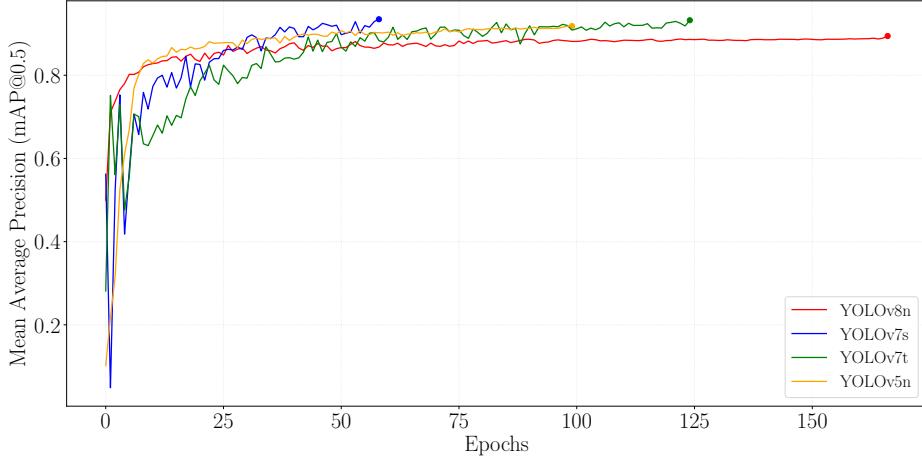
being asymmetric, and no class balancing process was carried out, under the assumption that the data set is wide and diverse enough. The dataset can be viewed and downloaded at the link in Appendix D.6, and the complete list of classes is in Table 2.

**Table 2:** List of classes from the public indoor signage image set

Image	Name	Description of the signage class
	aeroplane symbol	Signage containing an airplane symbol to indicate the location of flight arrival or departure gates.
	baggage claim	Signage indicating the direction to the baggage claim area.
	bathrooms, restaurants	Signage indicating the direction to bathrooms and places to eat, respectively.
	danger electricity	Signage indicating areas that present danger due to electrical shock.
	emergency exit	Signage indicating the direction to emergency exits.
	extinguisher symbol, fire extinguisher	Signage containing a fire extinguisher symbol, and a real fire extinguisher, respectively.
	handicapped symbol	Signage containing the international symbol of accessibility, used to indicate the direction to areas accessible to people with disabilities.
	up arrow, down arrow, left arrow, right arrow	Signage containing an arrow pointing up, down, left, or right, used to indicate the direction of the contextual information. These vary in thickness, color, and shape.

#### 4.2.2 Training of YOLO Models for Signage Detection

Training of the YOLO models was performed using a 40 GB NVIDIA A100 SXM4 graphics card provided by Google Colab. For each model, specific Colab notebooks were created and Google Drive was mounted to store the signage dataset files and the YOLO source code. The `data.yaml` and `train.py` files were modified to configure the file paths, number of classes, class names, and to set between 100 and 300 training epochs, stopping the process when the mAP validation accuracy exceeded 90%. Upon completion of each training, all relevant results were saved, such as confusion matrices, mAP@.5 accuracy data, and PyTorch (.pt) files with the final weights of each model. Figure 6 shows a graph of the accuracy of the YOLO models during the training of the signage dataset across the epochs, while Table 3 presents the final accuracy results by class.



**Fig. 6:** Accuracy of YOLO models during training on the signage dataset: YOLOv5n, YOLOv7t, YOLOv7s, and YOLOv8n.

**Table 3:** Accuracy results of trained models

Clases	Labels	mAP@.5 YOLOv5n	mAP@.5 YOLOv7t	mAP@.5 YOLOv7s	mAP@.5 YOLOv8n
airplane symbol	1380	87.1%	86.5%	90.3%	90.1%
baggage claim	166	94.1%	89.5%	91.6%	90.6%
bathrooms	310	91.7%	91.6%	91.6%	94.2%
danger	173	99.5%	97.0%	99.6%	95.0%
electricity					
down arrow	896	91.5%	87.4%	92.3%	83.1%
emergency exit	690	95.8%	94.0%	97.1%	95.0%
extinguisher symbol	123	89.3%	71.7%	87.5%	75.5%
fire					
extinguisher	192	88.2%	81.3%	83.5%	85.5%
handicapped symbol	1373	97.0%	94.0%	96.5%	97.1%
left arrow	1048	92.5%	92.3%	95.6%	92.7%
right arrow	977	86.3%	94.7%	88.8%	75.9%
restaurants	140	90.4%	89.0%	94.2%	90.7%
up arrow	1599	90.2%	91.5%	92.5%	90.8%
<b>Total</b>	8620	9.18%	89.3%	92.4%	88.9%

As can be seen in Table 3, the imbalance of the number of labels per class in the created dataset did not negatively affect the performance of the models, as no overfitting towards any specific class is evident. This is reflected in the consistency of the accuracy metrics across all classes, suggesting that the models have achieved good

1 performance in detecting signage in indoor public spaces, both for each individual  
 2 class and overall.

3 On the other hand, all the confusion matrices generated with the test dataset for  
 4 each of the trained models were analyzed, which can be found in section C. It could be  
 5 observed that all confusion matrices present a predominantly high main diagonal with  
 6 high values, reflecting a very high true positive rate compared to the false positive rate.  
 7 This indicates that the models were able to correctly classify the input images without  
 8 confounding the object classes. To illustrate this more clearly, some test images were  
 9 evaluated with the YOLOv7t model, and the results shown in Fig:TestResults were  
 10 obtained.



25 **Fig. 7:** Signal recognition model evaluation with some test images

### 30 4.3 Performance of neural models in the prototype

31 To perform the performance test described in the 3.3.2 section of the methodology,  
 32 the B.2 program was created to evaluate and compare the performance of different  
 33 YOLO models in the context of signaling and hand detection on an OAK-D device,  
 34 and to analyze the impact of performing hand tracking on the system's VPU or CPU.

35 The results of the performance tests are shown in the table  
 36 Öreftab:FrameRateTests. Where the first column indicates the model used, the sec-  
 37 ond, third and fourth columns represent the average frame rate of the predictions  
 38 of the YOLO models used different distributions for neural network processing. In  
 39 the first configuration, each YOLO model is processed individually along with the  
 40 generation of depth maps on the VPU of the OAK-D device. The second configura-  
 41 tion is similar to the previous one, but the Palm Detection and Hand Landmarks  
 42 model processing is also incorporated into the OAK-D device's OAK-D. And the  
 43 third configuration uses distributed neural network processing, where the YOLO  
 44 models are processed on the VPU of the OAK-D device, and the Palm Detection  
 45 and Hand Landmarks models are processed on the CPU of the Raspberry Pi 4 using  
 46

the MediaPipe API. Finally, the fifth column represents the percentage of successful detections of the test object, averaged over the different processing configurations.

**Table 4:** System tests using different processing configurations

Model for detecting public signage	Without hand detection	With hand detection on the <b>VPU</b>	With hand detection on the <b>CPU</b>	Successful detection rate ( <b>TP</b> )
YOLOv8n	13.84 fps	4.82 fps	9.26 fps	87.87 %
YOLOv7t	13.85 fps	5.45 fps	9.31 fps	84.84 %
YOLOv7s	1.19 fps	0.73 fps	1.11 fps	95.22 %
YOLOv5n	14.36 fps	5.35 fps	9.42 fps	78.41 %
Average	10.81 fps	4.09 fps	7.28 fps	86.09 %

It can be seen in the last row of Table 4 that the system performance was significantly better without hand detection, with an overall average of 10.81 fps, compared to the system configurations that include hand detection. This was expected, as hand detection involves the processing of two additional neural networks, which increases the computational load of the system. As for system configurations that incorporate hand detection, the overall average for the configuration with distributed processing reaches 7.28 fps, while the configuration with processing exclusively on the **VPU** gets 4.09 fps. Therefore, it can be said that the distributed processing of neural networks between the **VPU** and the **CPU** performs better than the processing of all neural networks exclusively on the **VPU**.

Particularly, the YOLOv5n model was the only one that presented failures in the detections of the test object, with an average of 20.34% false positives  $\overline{FP}$ . As for the average of successful detections  $\overline{TP}$ , the YOLOv7s model obtained the best result with 95.22% hits, however, this model had a frame rate far below the rest of the evaluated models. This last result is a consequence of the 415 abstraction layers that the small version of the YOLOv7 model has, compared to the 255 abstraction layers of the nano and tiny versions of the YOLO models. Therefore, it can be said that the increase in the percentage of successful detections due to the extra 160 abstraction layers of the YOLOv7s model, does not compensate for the decrease in the frame rate. Regarding the YOLOv8n and YOLOv7t models, it can be said that they present a balance between frame rates and successful detections, both models offer overall good performance on the **OAK-D** device.

Although the YOLOv7t model has a slightly lower percentage of successful detections than the YOLOv8n, this model was chosen to continue with the prototype testing due to its higher frame rate and lower computational resource consumption. The YOLOv7t offers an appropriate balance between real-time performance and accuracy in detections, which results in a more fluid and effective operation of the overall system.

#### 4.4 Thermal behavior of the system

In this section, the results and final details are described on how the experiments were carried out to compare and evaluate the thermal behavior of the system in response to the use of distributed neural network processing, as described in section 3.3.5 of the methodology. To achieve this, two independent experiments were carried out to digitally store the system's temperature during an hour of its operation, using the YOLOv7t model in common for processing the sign detections and the generation of depth maps on the **OAK-D** device's **VPU**.

A program was created, available in section B.3 of the appendix, to register the system's temperatures at three different points every second over a period of approximately one hour. To measure the temperature of the **OAK-D** device's heat sink, a MAX6675 thermocouple was screwed on the back, which was connected to the Raspberry Pi 4 as shown in Fig. 3 for temperature data transmission using the SPI protocol. In addition, a precision thermometer was used to adjust the digital temperature data, due to a constant discrepancy of approximately 5°C between the actual temperature and the temperature measured by the thermocouple. The **VPU**'s temperature was obtained by previously configuring a **SystemLogger** output node on the **OAK-D** device, and the **CPU**'s temperature was obtained through the output of the native Raspberry Pi system command `vcgencmd measure_temp`.

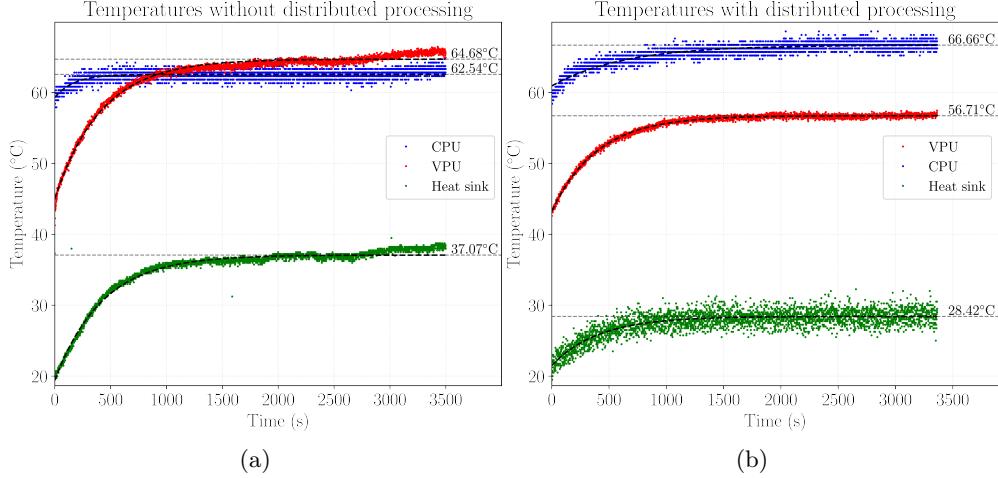
Subsequently, the temperature data was analyzed, and it was determined that the exponential growth described in equation (1) was the best curve fit for the increase in system temperatures as a function of time, with a coefficient of determination close to unity in most cases, and a standard deviation of 3°C at most in the case of the dispersion of data measured with the thermocouple.

$$T_{(t)} = T_{\infty} - (T_{\infty} - T_0)e^{-\frac{t}{\tau}} \quad (1)$$

Where  $T_{(t)}$  is the temperature at time  $t$ , and the constants  $T_0$ ,  $T_{\infty}$ , and  $\tau$  represent the initial temperature, the final temperature in a steady thermal state, and the first-order time constant of the thermal system, respectively.

Next, the graph shown in Fig. 8 was created, overlaying the temperature data and their corresponding curve fits, with the aim of comparing the thermal behavior of the system when it is configured both with the implementation of distributed neural network processing and without it. In the configuration with distributed processing, the **Palm Detection** and **Hand Landmarks** models run on the Raspberry Pi 4's **CPU** to detect the user's hand, while the YOLOv7 model is processed on the **OAK-D** device's **VPU** for the detection of public indoor signs. On the other hand, in the configuration without processing distribution, all the mentioned models run on the **OAK-D** device's **VPU**.

On the right side of the graphs in Fig. 8, projections of the system temperatures at steady thermal state  $T_{\infty}$  are included, obtained from the corresponding exponential curve fits, according to equation 1. It can be seen that the system takes approximately 25 minutes (equivalent to three times the time constant,  $3\tau \approx 1500[\text{s}]$ ) for the temperatures to reach the  $T_{\infty}$  value. When the system employs distributed neural network processing, the  $T_{\infty}$  temperature of the **CPU** increases by 4.12°C, as the **Palm Detection** and **Hand Landmarks** models are run on this computing unit. On



**Fig. 8:** System temperature measurements during operation: (a) System temperatures without distributed processing. (b) System temperatures with distributed processing

the other hand, the  $T_\infty$  temperature of the **VPU** notably decreases by  $7.97^\circ\text{C}$  as it does not require the execution of these models on that unit, resulting in a reduction of the heatsink's  $T_\infty$  temperature, from  $37.07^\circ\text{C}$  to  $28.42^\circ\text{C}$ .

Given that the ergonomic temperature limits for portable devices are  $41^\circ\text{C}$  for aluminum surfaces, and that the maximum operating temperature of the **OAK-D** device's **VPU** is  $105^\circ\text{C}$ , it can be concluded that the system is within safe temperature limits in both configurations. However, the configuration with distributed neural network processing results in lower temperatures for both the **VPU** and its heatsink, which is preferable from the point of view of user comfort and device lifespan. It is important to note that, although the system is within safe limits, continuous temperature monitoring is recommended during its operation to avoid potential overheating issues.

Based on the results obtained in this study, it can be concluded that it is not necessary to implement an additional cooling system for the **OAK-D** device in the configuration with distributed neural network processing. However, it is suggested to evaluate the system's thermal behavior under different environmental and usage conditions to ensure its proper functioning in a wide range of scenarios.

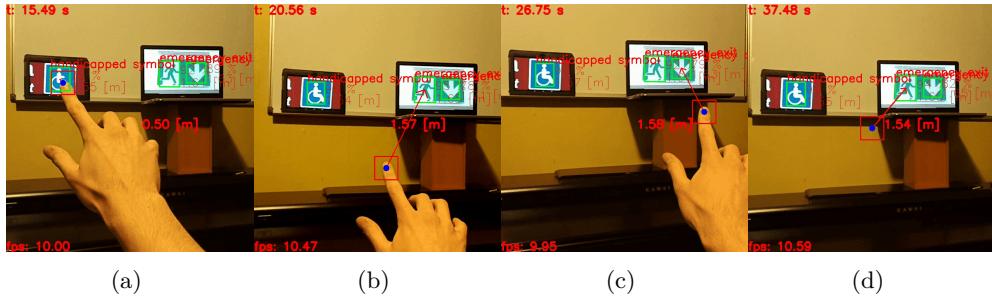
## 4.5 System Operation Measurements

Next, a set of tests and figures are shown to define the output events of the system under different situations, whose experimentation can be found in a YouTube video playlist available in section D.7 of the appendix.

### 4.5.1 Signage detection with user hand perturbations

Fig. 9 shows a sequence of video frames captured by the **OAK-D** sensor's **RGB** camera, in which vectors can be seen pointing to the detection closest to the reference point. In

figures 9a, 9b and 9c it can be seen that the user controls the reference point with the tip of their index finger to find the closest detection, while in Fig. 9d the center of the frame is used as a pre-established reference point so that the user's head movement is used to find the closest detection.



**Fig. 9:** Interaction with signs using the index finger and the head: (a) finger perturbation on the emergency exit sign, (b) control of the reference point with the index finger on the emergency exit sign, (c) control of the reference point with the index finger on the downward arrow sign, and (d) use of the center of the frame as a pre-established reference point to find the closest detection through head movement.

In this test, the system was configured with the distributed network processing procedure described in Section 4.4. An average frame rate of 9.83 fps was recorded during the detection of signage and the user's finger, which was used to disturb the detection of the signage. As shown in Figure 9a, the user's finger disturbs the detection of the emergency exit sign, however, the system is able to detect the signage correctly. In this scenario, it was observed that the system is capable of detecting signage with a partial obstruction of up to 25% of the signage by the user's finger with a detection confidence in the range of 66% to 72%, while detections that presented no disturbance reach a prediction confidence of 97%, as shown in Figures 9b and 9c. Unfortunately, detections with a 50% disturbance that exceeded the 60% minimum confidence were not observed.

In summary, the system demonstrates effective capability to detect signage in real-time, even with partial obstructions caused by the user's hand. The 25% resilience against disturbances and the frame rate of 9.83 fps allow for fluid and reliable user interaction. Furthermore, the system's ability to adapt to different control techniques, such as the use of the index finger or head movement, suggests flexibility in its application for different contexts and user preferences. Although the tests presented in this study are promising, additional studies could be conducted to explore the system's performance in different environments and with varying levels of visual noise.

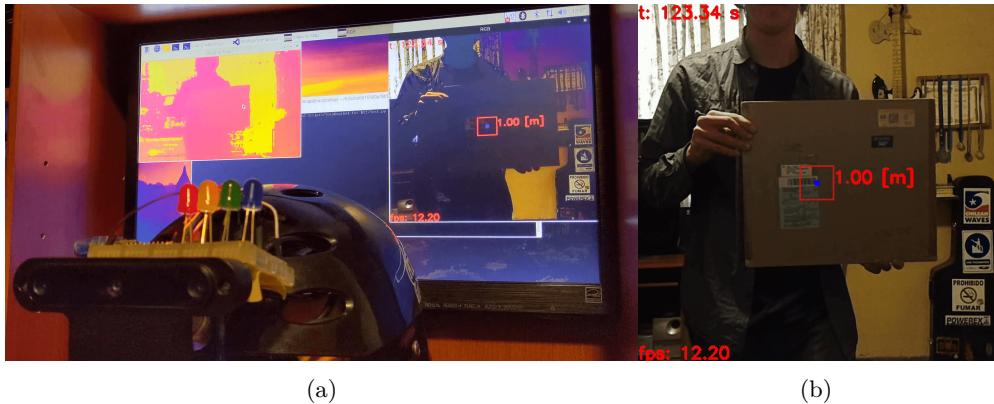
#### 4.5.2 Communication Interface Tests

To evaluate the behavior of the communication interface, a two-stage test was conducted, the results of which are presented in Sections 4.5.2 and 4.5.2. The first stage

involves recording the obstacle alerts generated on the communication interface, varying the distance of a test obstacle measured by the **OAK-D** sensor in the central ROI of the depth map. The second stage involves recording the number of pixels between the virtual reference and the center of the nearest signage detection in the frames captured by the **RGB** camera. It is important to mention that in these tests, the vibrotactile actuators {UP, DOWN, LEFT, RIGHT} were replaced with color LEDs {GREEN, RED, BLUE, YELLOW} respectively, in order to record the haptic operation of the communication interface using an external camera. In addition, for this test, the system was configured without the processing of hand detections, keeping the virtual reference fixed in the center of the frame, so as not to affect the tracking of the nearest detection. As a result of this configuration, the average response time of detections was 76.12 ms, which is equivalent to a frame rate of 14.43 fps. The programs and the video of this test are found in the appendix, in Sections B.4 and D.7, respectively.

#### **Test of proximity alerts for obstacles**

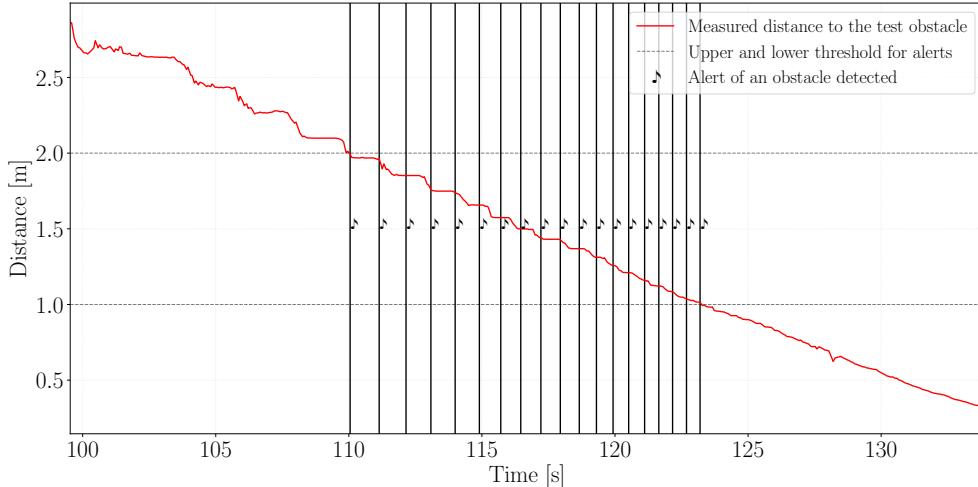
Figure 10a shows the user graphical interface (GUI) of the Raspberry Pi on the monitor, including the depth map of the **OAK-D** sensor in the upper left corner. Figure 10b presents the output from the **OAK-D** sensor's **RGB** camera, with the central ROI in red and a test obstacle 1 meter away.



**Fig. 10:** Obstacle proximity alerts test setup: (a) Monitor connected to Raspberry Pi. (b) Test obstacle.

Figure 11 shows a plot representing the variation of the distance, measured in meters, between the test obstacle and the sensor's stereo camera **OAK-D** over time. The obstacle was approached head-on to the camera at a nearly constant speed, always remaining within the central ROI of the depth map to maintain a continuous distance measurement. In the plot, vertical lines with a  $\downarrow$  symbol have been added at times when the communication interface buzzer emits 75 ms to 130 Hz tones as proximity alerts, becoming more frequent as the obstacle approaches. Horizontal asymptotes have also

been drawn to represent the limits of the obstacle alert threshold, set between one and two meters away.



**Fig. 11:** Test obstacle proximity alerts

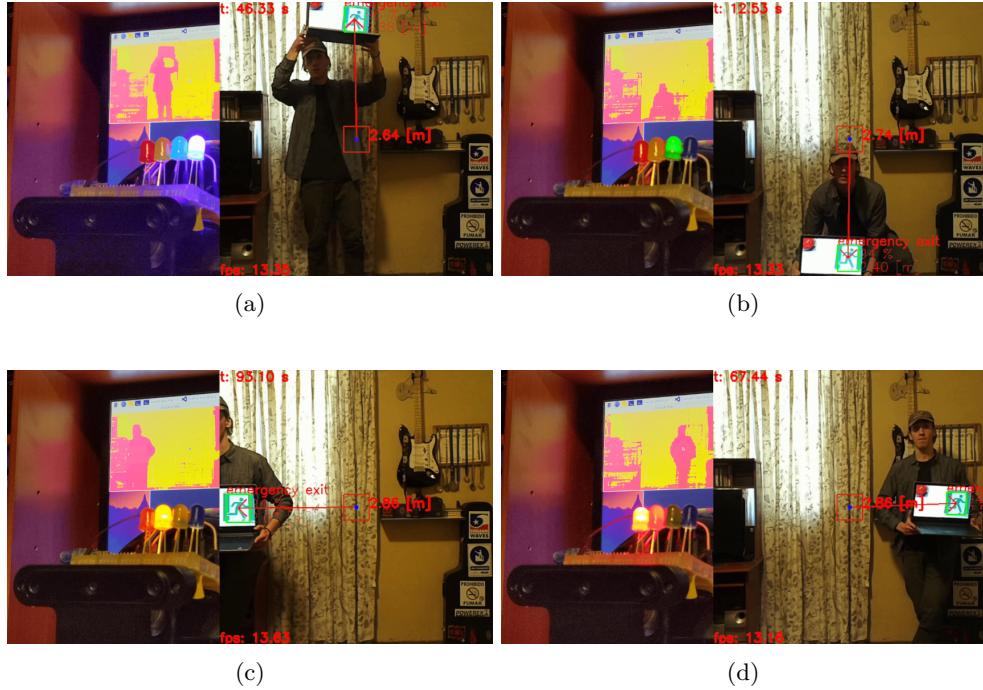
During the test, no errors in communication or distance measurement were recorded. Figure 11 evidences the proper and consistent behavior of the communication interface in relation to obstacle alerts. Proximity alerts are correctly issued and become more frequent as the obstacle approaches the sensor, indicating an effective system response to changes in the distance to the obstacle. The absence of communication and measurement errors demonstrates the robustness and reliability of the communication interface in this context.

In summary, the results obtained in this test suggest that the communication interface is capable of providing accurate and timely proximity alerts, thus contributing to better system performance in the detection and prevention of collisions with obstacles.

#### **Test of tracking the nearest detection**

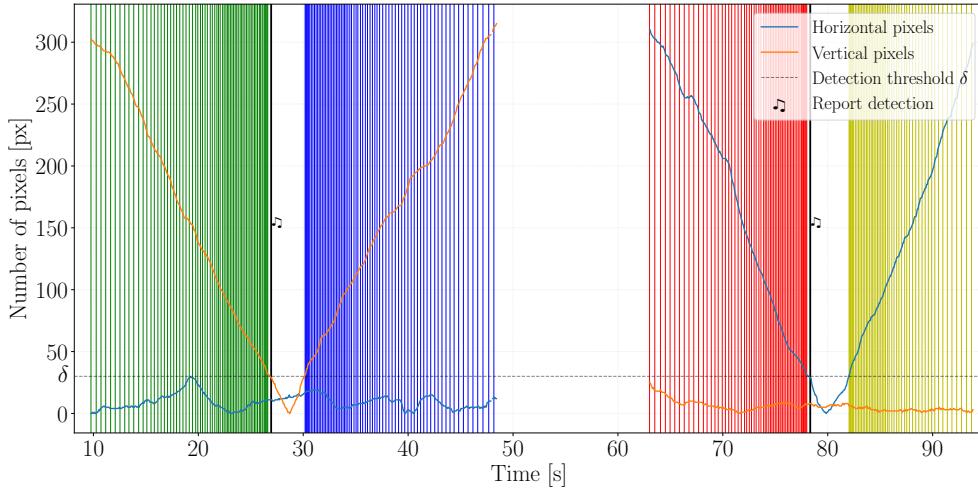
The objective of this test is to demonstrate the operation of tracking the detection closest to the virtual reference, and to evaluate the effectiveness of the haptic and audible indications of the communication interface. To this end, the frames captured by the OAK-D sensor's RGB camera were recorded, while a test signage was moved, in this case, an emergency exit symbol was used on a laptop screen. The test signage was moved from bottom to top and from right to left passing through the central ROI, to generate a variation of the distance between the virtual reference and the nearest detection over time, while the activity of the vibrators and the audio output of the communication interface were recorded, using the program B.4 in the appendix. Figure 12 shows a sequence of four vertically split images to better visualize how this

procedure was carried out in parallel with the feedback from the communication interface. In each of the images, a lit color LED can be seen on the left side, representing the activation of a vibrator {UP, DOWN, LEFT, RIGHT} present in Figure 2a, and on the right side of the image, the frame with an arrow pointing to the detection closest to the virtual reference can be seen.



**Fig. 12:** Representation of the activation of the vibrators in the communication interface: (a) Activation of the DOWN vibrator. (b) Activation of the UP vibrator. (c) Activation of the RIGHT vibrator. (d) Activation of the LEFT vibrator.

In Fig. 13, recordings of the communication interface activity over time are shown, where discontinuous segments of light-blue and orange curves can be observed. These represent the quantity of horizontal and vertical pixels between the center position of the nearest detection and the reference as time progresses, respectively. Alongside these curve segments, vertical bands of blue, green, yellow, and red can be seen, signifying moments when colored LEDs are activated. At the end of each color band set, there is a noticeable thick black stripe with a  $\blacktriangleright$  symbol, which represents the point in time when the Raspberry Pi sends an audio signal with a contextual TTS message mentioning the detected symbol class and its distance. It is important to note that throughout the duration of this test, no false detections were annexed to the test signal, and the system was able to detect the test signal at all times, except for a brief moment, reflected in a small discontinuity at the end of the first orange curve segment.



**Fig. 13:** Haptic indications of communication interface

It can be noted in the colored vertical bands of Fig. 13 that, while detecting the test signal, the frequency of vibrator activation increased as the number of pixels in the direction where the detection was furthest from the reference decreased. In this case, as the test signal moved from bottom to top and right to left, activating the DOWN, UP, RIGHT, and LEFT vibrators in that order, respectively. This demonstrates that the communication interface is capable of indicating to the user in real time when they are approaching or moving away from a signal detection, and in which directions they need to move the virtual reference to find the nearest detection.

On the other hand, the vibrators remained deactivated as long as the condition that the pixel count is below the detection threshold  $\delta$  of 30 pixels was met. At these times, black bands with the ♪ symbol can be observed, which only appear the first time the aforementioned condition is met. This indicates that the system does not repeat the same audio message in each iteration of the control loop. Additionally, the system does not overload the user with audio messages from all the detections that are in the field of vision of the RGB camera, nor with repetitive audio messages from the same detection.

## 5 Discussion of Results

Following the research and methodology of chapters 2 and 3, a functional prototype of the system was constructed with a lightweight and portable design, integrating specific hardware and software components for AI processing. To achieve this, a sufficiently large and diverse dataset of public signals was created, which was crucial for training different versions of YOLO with accuracies close to 0.9 mAP without presenting overfitting on a particular class or confusion between classes. This dataset, along with the trained detection models and all the software tools used for the development of this prototype, have been made available to the community in a freely accessible GitHub

repository, contributing to the scientific and technological community in advancing the creation of solutions to improve the quality of life and independence of people with blindness and visual impairment.

Following the successful implementation of the trained neural models on the **OAK-D** device and the methodological tests, it was discovered that distributing the models used for hand and signal detection among different processing units increased their capacity to perform simultaneous predictions at 9.31 fps and, in turn, improved the thermal behavior of the system without the need for an additional cooling system, implying a longer battery life, increasing the portability and usability of the device. On the other hand, the system's operation records reflected robust characteristics of the prototype for detecting public signals with the user's own finger obstruction, maintaining acceptable confidence ranges between 66% to 72%, and 97% without obstructions. In addition, the reliability of the system's communication interface was confirmed, as there were no transmission errors of bytes sent to the microcontroller for the issuance of haptic and audible feedback. Finally, by providing this information to the user at a variable frequency, which depends on the proximity to obstacles and the position of the nearest detection to the virtual reference, the system's potential to guide and direct a person with **BVI** in real time was demonstrated, without overloading the user with obstacle alerts or repetitive audio messages from all possible detections in the field of vision of the **RGB** camera.

## 6 Conclusions

In conclusion, it can be stated that the **AI**-based **ETA** system proposed in this work not only adopts the characteristic of a guide dog that guides and directs any user with **BVI**, but also overcomes several technical challenges through the use of advanced computer vision technologies, object detection techniques, and the implementation of an effective communication interface. This enhances the user's orientation and movement in public interiors with improved safety and confidence.

The development of the system demonstrates its viability and efficacy through a series of evaluations that have measured its accuracy, processing speed, robustness, and thermal behavior, thereby meeting the overall objective.

The developed system is capable of alerting the user about potential obstacles and guiding them to locate signals in real time, using the movement of their index finger and/or head. This is achieved through the integration of stereoscopic vision, neural models for hand and signal detection, and a communication interface that provides the user with feedback through haptic and audible signals.

These achievements mark a significant step forward in providing technological solutions to improve the quality of life and independence of people with blindness and visual impairment, opening up new areas of opportunity for future research and development in this field.

## 7 Limitations and Future Research

Despite the significant achievements made with the developed prototype, some limitations persist, which present areas of opportunity for future research and improvements,

such as the system's robustness under challenging lighting conditions like darkness. This could be improved by incorporating additional sensors, such as thermal cameras or **LiDAR** sensors. Additionally, the integration of computer vision algorithms, for instance, sign segmentation and text detection with **OCR** on them, could enrich the information provided to the user, expanding the system's utility in different public spaces.

Adapting the developed prototype for use in sports activities would be an attractive line of future research, as it improves the quality of life for people with **BVI**, and contributes to their health by allowing them to engage in sports activities with more safety and confidence. Another interesting modification would be the selection of different easy-to-process models oriented for use in various environments, through the detection of user hand gestures, since the hardware limitations of a portable device do not allow the processing of several models simultaneously or a complex model that requires a large amount of computational resources.

Lastly, the inclusion of real users during the testing and evaluation phases is essential. Their contributions will enable the optimization of the user experience and better adapt the system to the needs of the target audience. Therefore, future research should plan usability tests with people with **BVI**, to ensure that the developed system effectively meets their needs.

**Supplementary information.** The study is accompanied by a GitHub repository, which contains all the programming codes, model configurations, and other relevant software elements crucial for this research. The repository can be accessed at: <https://github.com/NicoGitSoft/BVI>.

Key programming scripts of relevance include: - DepthYoloHandTracker class: <https://github.com/NicoGitSoft/BVI/blob/main/Scripts/Utilities.py> - System Performance Evaluation: <https://github.com/NicoGitSoft/BVI/blob/main/Scripts/FrameRateTest.py> - Thermal Behavior Evaluation: <https://github.com/NicoGitSoft/BVI/blob/main/Scripts/TemperatureTest.py> ... [and so on for other key scripts].

Full-resolution confusion matrices for all YOLO model trainings can be found within the repository or as supplementary figures.

**Acknowledgments.** We are grateful for the support from Dicyt Project 062117SG, Vicerrectoría de Investigación, Innovación y Creación, FONDEF ID21I10191, and STIC-AmSud AMSUD220026.

## Declarations

- **Funding:** None.
- **Conflict of Interest/Competing Interests:** None.
- **Ethics Approval:** Not applicable.
- **Consent to Participate:** Not applicable.
- **Consent for Publication:** Not applicable.
- **Data and Materials Availability:** Available in the appendix (repository).
- **Code Availability:** Available in the appendix (repository).
- **Authors' Contributions:**

- Nicolas Ibanez: Design, implementation, and writing of the paper.  
 – Ismael Soto: Proofreading and editing of the paper.

## List of Acronyms

AFB	American Foundation for the Blind	1
AI	Artificial Intelligence	2–4, 7, 9, 29, 30
BVI	Blindness and Visual Impairment	1–3, 7, 8, 30, 31
CPU	Central Processing Unit	11, 21–23, 33, 34
CV	Computer Vision	4, 8–10
DL	Deep Learning	8
ETA	Electronic Travel Aid	2–4, 7, 8, 30
FN	False Negative	10
FP	False Positive	10
GPU	Graphics Processing Unit	36
IoU	Intersection over Union	10
LiDAR	Light Detection and Ranging	31
mAP	mean Average Precision	10
NFB	National Federation of the Blind	1
OAK-D	OpenCV AI Kit Depth	2, 9, 11, 13–18, 21–24, 26, 27, 30, 33, 34, 38, 39
OCR	Optical Character Recognition	4, 31
R-CNN	Region-based Convolutional Neural Network	10
RGB	Red Green Blue	24, 26, 27, 29, 30
ROI	Region of Interest	15
SPI	Serial Peripheral Interface	14
TP	True Positive	10
TTS	Text-to-Speech	4, 7, 9, 15, 28
UART	Universal Asynchronous Receiver-Transmitter	15
VPU	Vision Processing Unit	2, 11, 17, 21–24, 33, 34
YOLO	You Only Look Once	9, 10, 17–21, 33, 36

## Appendix A Repository

For the development of this work, a publicly accessible GitHub repository has been created to store all the software elements that were implemented for the development of the proposed system. Below is a description of the repository organization, available at the following link: <https://github.com/NicoGitSoft/BVI> [70]

The **Scripts** folder contains all the programming codes used, the **Arduino** folder contains the programs related to the Atmega328P microcontroller, and the **Notebooks** folder contains the Colab notebooks used for training the YOLOv5n, YOLOv7t, YOLOv7s and YOLOv8n models with the set of images of public indoor signage present in the **Dataset** folder. The **Models/Songs** folder contains the files with the

weights already trained for the **YOLO** models in PyTorch, OpenVINO and ONNX format, together with their respective configurations in JSON and XML format. Finally, in the folder **Models/Hands**, there are the Palm Detection and HandLandmark models in OpenVINO (.blob) format for direct use in the **OAK-D** device.

## Appendix B Programming codes

In this appendix, only those code fragments that are of greater relevance for the development of the system proposed in this *Memoria* have been included. The codes used in the training of the **YOLO** models, as well as the custom libraries, utilities and classes developed, have not been included in this appendix in order to avoid excessive extension and to guarantee a better understanding and clarity in the exposition of the content. However, these elements can be consulted in the GitHub repository associated to this report, mentioned in [A](#).

### B.1 Class created for configuring DepthAI nodes

The created class **DepthYoloHandTracker** allows configuring and controlling a **OAK-D** device to perform hand and object detection tasks using YOLO models. The class is responsible for creating and managing a pipeline that uses monochrome and color cameras and neural network nodes to process the captured images. It also allows real-time hand and signal detection, and temperature measurement on the VPU. In addition, different parameters, such as models and configurations, can be modified to adapt the class to different use cases. Available at: <https://github.com/NicoGitSoft/BVI/blob/main/Scripts/Utilities.py>

### B.2 System performance evaluation

This program was created to evaluate and compare the performance of different **YOLO** models in the context of signaling and hand detection on an **OAK-D** device, and to analyze the impact of performing hand tracking on the system's **VPU** or **CPU**. It uses methods of the **DepthYoloHandTracker** class to change the **YOLO** models on the **OAK-D** device and creates a comparison table of the results. The performance of the system with hand detection processing on the **CPU** and on the **VPU** is measured, as well as the performance of the system without hand detection processing. The results are displayed in a table that includes performance metrics in terms of FPS and percentages of successful, missed and undetected detections. Available at: <https://github.com/NicoGitSoft/BVI/blob/main/Scripts/FrameRateTest.py>

### B.3 Evaluation of the thermal performance of the system

This program was created to evaluate the thermal behavior of the system during a test of approximately one hour. It uses the **DepthYoloHandTracker** class with the **temperature\_sensing=True** method to measure the temperature of the **VPU** on the **OAK-D** sensor and the **max6675** custom library to measure the temperature of the heat sink with the thermocouple connected to the Raspberry Pi via SPI communication. In addition, the temperature of the Raspberry Pi's **CPU** is measured using

the "vcgencmd measure\_temp" command. The code loads the SingsYOLOv7t model and its configuration, initializes the OAK-D device with specific parameters and configures the pins and objects needed for communication with the MAX6675 sensor. In the main loop, the instantaneous temperature of the VPU is obtained and the measured temperatures of the VPU, the CPU and the MAX6675 sensor are stored every second. The temperature values are displayed on the console with an accuracy of 2 decimal places. At the end of the test, the time and temperature samples are saved in a CSV file. Available at: <https://github.com/NicoGitSoft/BVI/blob/main/Scripts/TemperatureTest.py>

#### B.4 Evaluation of the user communication interface

The following programs, B.4.1 and B.4.2, interact via serial communication to perform and save events from the user communication interface for evaluation. The B.4.1 program, running on the Atmega328P microcontroller, is responsible for controlling a buzzer and a haptic interface with four directional vibrators, receiving commands from the B.4.2 program, running on the Raspberry Pi 4. The latter program, B.4.2, is designed to detect objects in real time and provide haptic and auditory feedback to the user based on the position and distance of the detected objects.

#### B.4.1 Program on Atmega328P microcontroller

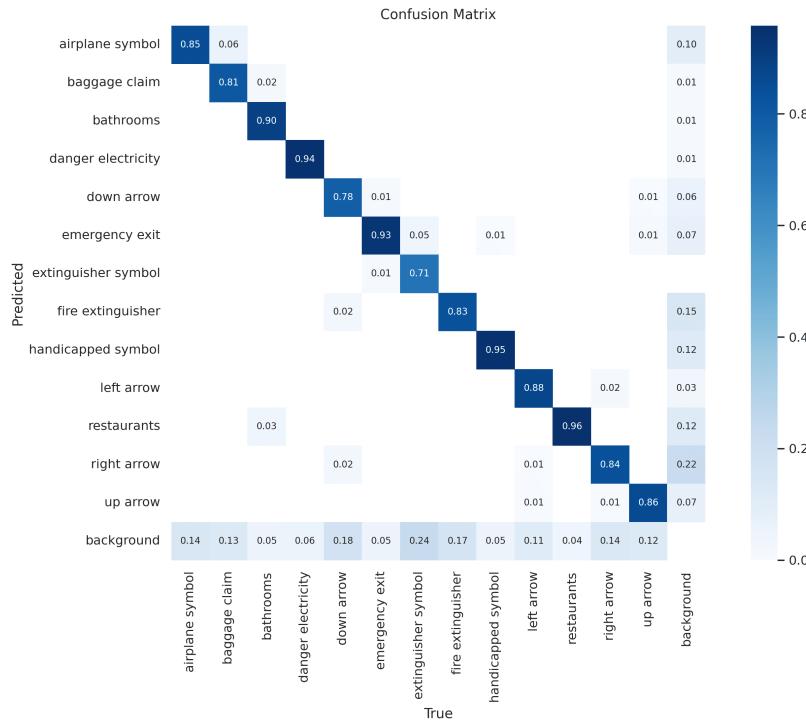
The code is a program in Arduino language that uses serial communication to receive data and control a buzzer and a haptic interface with four directional vibrators. It declares constants and global variables to define the connection pins of the devices and the characteristics of the signals sent, such as the frequency and duration of the musical notes. In the configuration process, the serial communication is initialized to 9600 bps, the connection pins of the devices are defined as outputs and the duration of the musical notes is set. In the main loop, if data is available on the serial port, a byte is read and the buzzer or directional vibrators are activated according to the value received, using the "tone" function to generate the signal with the specified frequency and duration. The commands sent by the serial port control the direction of the haptic vibration and the activation of the buzzer. Available at: <https://github.com/NicoGitSoft/BVI/blob/main/Scripts/Arduino/Atmega328P/Atmega328P.ino>

#### B.4.2 Program on Raspberry Pi 4 microprocessor

This code is designed to detect objects in real time and provide haptic and auditory feedback to the user based on the position and distance of the detected objects. This program sets reference points based on the user's hand position and the distance from the wrist to the camera, and provides haptic feedback based on the position of the object closest to the reference point. A buzzer is also used to emit sounds with increasing frequency as an object approaches a predefined central region of interest (ROI). Haptic feedback and detected object tags are stored, along with temperature information from the sensor chip and Raspberry Pi. In addition, results are displayed and FPS is calculated in real time. The program can be terminated by pressing specific

keys, and the collected data is saved to a .mat file upon completion. Available at:  
[https://github.com/NicoGitSoft/BVI/blob/main/Scripts/BVI\\_MAIN.py](https://github.com/NicoGitSoft/BVI/blob/main/Scripts/BVI_MAIN.py)

## Appendix C Confusion matrices



**Fig. C1:** Confusion matrix of YOLOv8 nano training

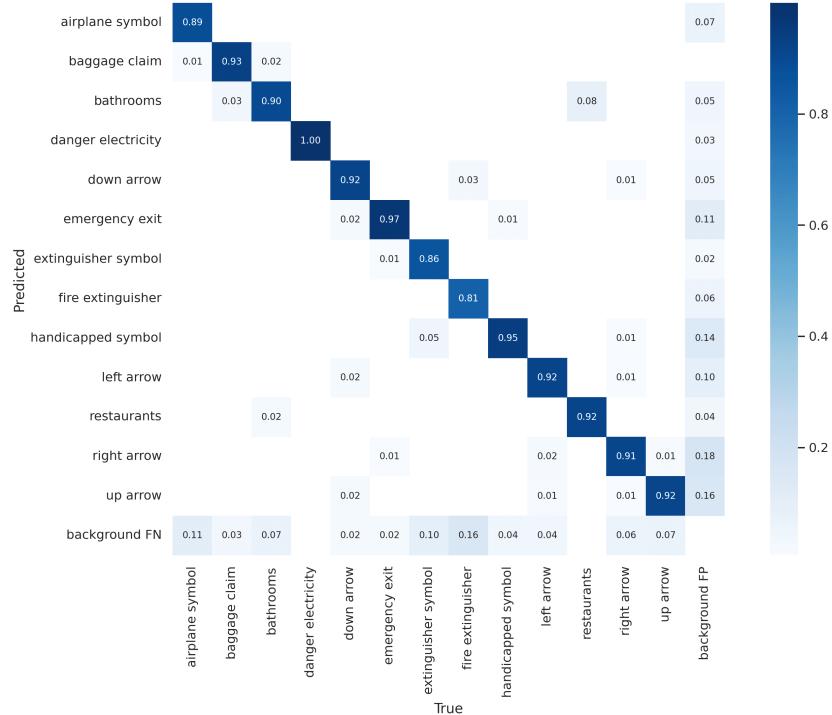
## Annex D

### D.1 YOLO Repositories

- Repositorio oficial de YOLOv5: <https://github.com/ultralytics/yolov5>
- Repositorio oficial de YOLOv7: <https://github.com/WongKinYiu/yolov7>
- Repositorio oficial de YOLOv8: <https://github.com/ultralytics/yolov8>

### D.2 DepthAI Documentation

DepthAI is a spatial platform that allows robots and computers to perceive the world as a human does: what objects or features they are and where they are in the physical



**Fig. C2:** Confusion Matrix of YOLOv7 small Training

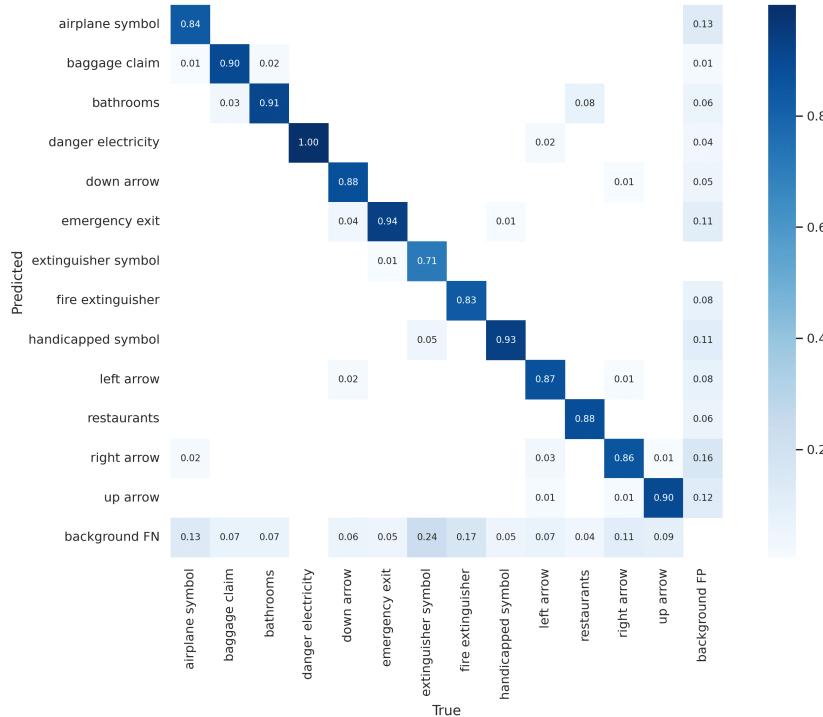
world. It focuses on the combination of these 5 key features: Artificial Intelligence, Computer Vision, Depth Perception (Stereo, ToF), Performance (high resolution and FPS, multiple sensors) and Low Power Integrated Solution. For more information, the official DepthAI documentation is available at the following link: <https://docs.luxonis.com/en/latest/>

### D.3 Tools for converting and exporting YOLO models

The <https://tools.luxonis.com/> page is a Luxonis tool for automatically exporting YOLO models for OAK devices. In order to simplify the process of exporting the most popular object detectors, they developed this tool. You simply load the pre-trained model weights (.pt file) and it will compile an OpenVINO file (.blob extension) and another JSON configuration file, and then run the trained object detector on OAK devices with these files using the DepthAI API.

### D.4 Google Colab Platform

Google Colab is an online platform that allows you to write and run Python code in a web browser. It is a very useful tool for machine learning, data science and education in general. To access Google Colab and make use of a free GPU for training



**Fig. C3:** Confusion Matrix of YOLOv7 tiny training

machine learning models, the platform can be accessed at the following link: <https://colab.research.google.com/>

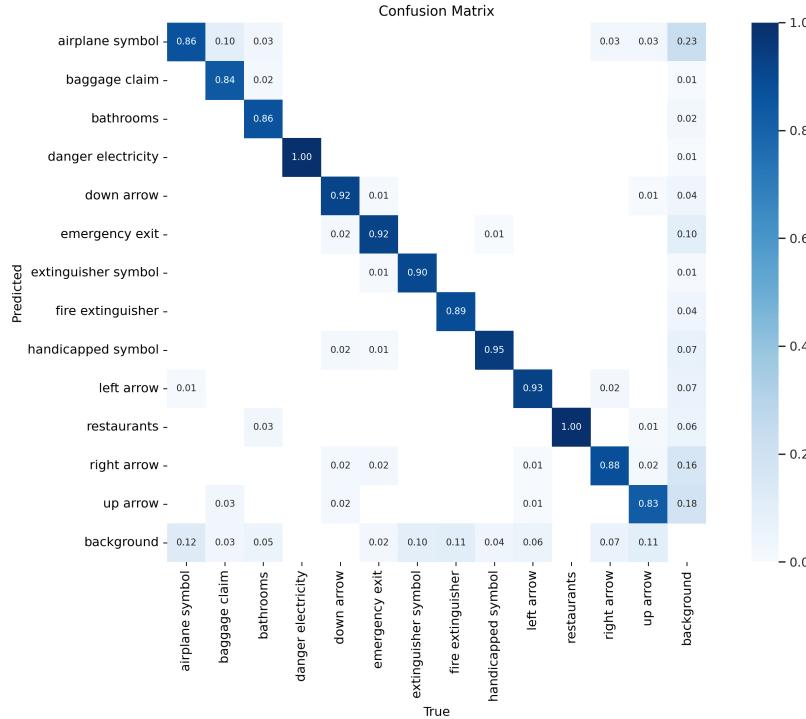
## D.5 Roboflow Platform

Roboflow is a platform that facilitates the creation of image datasets through image annotation, image labeling, data augmentation and dataset management. To access Roboflow you must create an account and log in at the following link: <https://roboflow.com/>.

In the D.6 section is the link to the set of public indoor signage images created in Roboflow for training the YOLOv5n, YOLOv7t, YOLOv7s and YOLOv8n models.

## D.6 Dataset of public indoor signage created in Roboflow

The set of public indoor signage images created in Roboflow for training the YOLOv5n, YOLOv7t, YOLOv7s and YOLOv8n models can be found at the following link: <https://app.roboflow.com/generic-signage/airports-and-subways/7>.



**Fig. C4:** Confusion matrix of YOLOv5 nano training

## D.7 Videos of tests performed

The videos corresponding to the tests performed with the prototype developed in this report are available in the links below, accompanied by a brief description of each one:

- Playlist of videos of tests performed: This is a complete playlist of all the videos of tests performed with the prototype developed in this report. The link to the playlist is as follows: [https://youtube.com/playlist?list=PLXpCoWH2IoQhF9UD4s4WAXlVEs\\_a9sLtz](https://youtube.com/playlist?list=PLXpCoWH2IoQhF9UD4s4WAXlVEs_a9sLtz)
- Testing of obstacle alerts and tracking of the closest detection to the reference point: This video corresponds to the experimentation described in sections 4.5.2 and 4.5.2. The objective of the video was to record both obstacle proximity alerts and haptic feedback to the user. It can be seen that the system is able to detect obstacles and track the closest signage to the landmark, providing haptic and audible feedback to the user when approaching the signage via colored LEDs indicating the direction of the haptic vibration. The link to the video is below: <https://www.youtube.com/watch?v=YB01FMMtBHI>
- Distance measurement range test with a OAK-D camera: This is a test attached to this report, which was performed to evaluate the distance measurement range of

1 the OAK-D camera. The link to the video is as follows: [https://youtu.be/Osb4\\_oEmE\\_o](https://youtu.be/Osb4_oEmE_o)  
2

- 3 • Signal detection test in an airport: This is a test attached to this report, in which  
4 the YOLOv8n model is evaluated with a video at the airport of Santiago de Chile.  
5 The Roboflow platform was used to modify in real time the parameters such as the  
6 confidence threshold and the IoU threshold in order to evaluate the performance of  
7 the model. The link to the video is as follows: <https://youtu.be/A9U7KaCw8K0>

8 **References**  
9

- 10 [1] Bourne RRA, Adelson JD, Flaxman S, Briant PS, Bottone M, Vos T, et al. Global  
11 Prevalence of Blindness and Distance and Near Vision Impairment in 2020:  
12 progress towards the Vision 2020 targets and what the future holds. Investigative  
13 Ophthalmology & Visual Science. 2020;61:2317–2317.
- 14 [2] Steinmetz JD, Bourne RR, Briant PS, Flaxman SR, Taylor HR, Jonas JB, et al.  
15 Causes of blindness and vision impairment in 2020 and trends over 30 years, and  
16 prevalence of avoidable blindness in relation to VISION 2020: the Right to Sight:  
17 an analysis for the Global Burden of Disease Study. The Lancet Global Health.  
18 2021;9(2):e144–e160.
- 19 [3] Evans JR, Fletcher A, Wormald R, Ng ESW, Stirling S, Smeeth L, et al. Preva-  
20 lence of visual impairment in people aged 75 years and older in Britain: results  
21 from the MRC trial of assessment and management of older people in the  
22 community. British journal of Ophthalmology. 2002;86:795 – 800.
- 23 [4] Thornett CEE, Brown AWS. Interfacing modern technology to disabled patients.  
24 In: IEE Colloquium on Engineering Design for the Disabled; 1988. p. 9/1–9/2.
- 25 [5] Smith P, Smith L. Artificial intelligence and disability: too much promise,  
26 yet too little substance? AI and Ethics. 2021;1. <https://doi.org/10.1007/s43681-020-00004-5>.
- 27 [6] Dakopoulos D, Bourbakis NG. Wearable Obstacle Avoidance Electronic Travel  
28 Aids for Blind: A Survey. IEEE Transactions on Systems, Man, and Cybernetics,  
29 Part C (Applications and Reviews). 2010;40(1):25–35. <https://doi.org/10.1109/TSMCC.2009.2021255>.
- 30 [7] The Americans with Disabilities Act (ADA). Guide to the ADA Accessibil-  
31 ity Standards; 2011. Available from: <https://www.access-board.gov/ada/guides/chapter-7-signs/>.
- 32 [8] Bourne RR, Taylor HR, Flaxman SR, Keeffe J, Leasher J, Naidoo K, et al.  
33 number of people blind or visually impaired by glaucoma worldwide and in world  
34 regions 1990–2010: a meta-analysis. PloS one. 2016;11(10):e0162229.

- [9] Burton MJ, Ramke J, Marques AP, Bourne RR, Congdon N, Jones I, et al. The Lancet global health Commission on global eye health: vision beyond 2020. *The Lancet Global Health*. 2021;9(4):e489–e551.
- [10] Khairallah M, Kahloun R, Bourne R, Limburg H, Flaxman SR, Jonas JB, et al. number of people blind or visually impaired by cataract worldwide and in world regions, 1990 to 2010. *Investigative ophthalmology & visual science*. 2015;56(11):6762–6769.
- [11] Flaxman SR, Bourne RRA, Resnikoff S, Ackland P, Braithwaite T, Cicinelli MV, et al. Global causes of blindness and distance vision impairment 1990-2020: a systematic review and meta-analysis. *LANCET GLOBAL HEALTH*. 2017 DEC;5(12):E1221–E1234. [https://doi.org/10.1016/S2214-109X\(17\)30393-5](https://doi.org/10.1016/S2214-109X(17)30393-5).
- [12] Stevens GA, White RA, Flaxman SR, Price H, Jonas JB, Keeffe J, et al. Global prevalence of vision impairment and blindness: magnitude and temporal trends, 1990-2010. *Ophthalmology*. 2013 Dec;120:2377–2384. <https://doi.org/10.1016/J.OPHTHA.2013.05.025>.
- [13] Furtado JM, Lanssing VC, Carter MJ, Milanese MF, Peña BN, Ghersi HA, et al. Causes of blindness and visual impairment in Latin America. *Survey of ophthalmology*. 2012 Mar-Apr;57(2):149–177. <https://doi.org/10.1016/j.survophthal.2011.07.002>.
- [14] Ministerio de Desarrollo Social y Familia. Informe de Desarrollo Social 2022. Ministerio de Desarrollo Social y Familia; 2022. Available from: <https://www.desarrollosocialyfamilia.gob.cl/storage/docs/ids/Informe-desarrollo-social-2022.pdf>.
- [15] THE WORLD BANK. Population ages 65 and above (% of total population) - Chile; 2022. Available from: <https://data.worldbank.org/indicator/SP.POP.65UP.TO.ZS?locations=CL>.
- [16] de Estadística IN. Adultos mayores en Chile: ¿Cuántos hay? ¿Dónde viven? ¿Y en qué trabajan?; 2022.
- [17] SENADIS. Segundo Estudio Nacional de la Discapacidad. Resultados Regionales para la Población Adulta. 2015;p. 9–10.
- [18] Subsecretaría de evaluación social de Chile - Ministerio de Desarrollo Social y Familia. Documento de resultados: Personas mayores, envejecimiento y cuidados; 2022. Available from: [http://observatorio.ministeriodesarrollosocial.gob.cl/storage/docs/grupos-poblacion/Documento\\_de\\_resultados\\_Personas\\_mayores\\_envejecimiento\\_y\\_cuidados\\_31.07.2020.pdf](http://observatorio.ministeriodesarrollosocial.gob.cl/storage/docs/grupos-poblacion/Documento_de_resultados_Personas_mayores_envejecimiento_y_cuidados_31.07.2020.pdf).
- [19] Gilbert CE, Canovas R, Kocksch de Canovas R, Foster A. Causes of Blindness and Severe Visual Impairment in Children in Chile. *Developmental Medicine and*

Child Neurology. 1994 Apr;36(4):326–33. <https://doi.org/10.1111/j.1469-8749.1994.tb11853.x>.

- [20] DeAngelis T. Optimizing tech for older adults. Monitor on Psychology. 2021;52(5):54.
- [21] Harris MT, Blocker KA, Rogers WA. Older Adults and Smart Technology: Facilitators and Barriers to Use. Frontiers in Computer Science. 2022 May;4. <https://doi.org/10.3389/fcomp.2022.835927>.
- [22] Kumar K, Champaty B, Uvanesh K, Chachan R, Pal K, Anis A. Development of an ultrasonic cane as a navigation aid for the blind people. In: 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT); 2014. p. 475–479.
- [23] Al-Fahoum AS, Al-Hmoud HB, Al-Fraihat AA. A smart infrared microcontroller-based blind guidance system. Active and Passive Electronic Components. 2013;2013.
- [24] Nada AA, Fakhr MA, Seddik AF. Assistive infrared sensor based smart stick for blind people. In: 2015 science and information conference (SAI). IEEE; 2015. p. 1149–1154.
- [25] F A, NADA A, A M, MASHALI S. Effective Fast Response Smart Stick for Blind People. In: Second International Conference on Advances in Bio-Informatics and Environmental Engineering - ICABEE 2015. Institute of Research Engineers and Doctors; 2015. p. 1–3. Available from: <https://doi.org/10.15224/978-1-63248-043-9-29>.
- [26] Alam UK, Rabby F, Islam M. Development of a Technical Device Named GPS Based Walking Stick for the Blind. Rajshahi University journal of Science and Engineering. 2015;43:73–80.
- [27] Nieto L, Padilla C, Barrios M. Design and implementation of elctronic aid to blind's cane. In: 2014 III international congress of engineering mechatronics and automation (CIIMA). IEEE; 2014. p. 1–4.
- [28] Kumar NA, Thangal YH, Beevi KS. IoT Enabled Navigation System for Blind. 2019 IEEE R10 Humanitarian Technology Conference (R10-HTC)(47129). 2019;p. 186–189. <https://doi.org/10.1109/R10-HTC47129.2019.9042483>.
- [29] Asha T, Krishnan JG, Sreelakshmi J, Jyothish K, Lavanya R. Haptic Hat for Visually Impaired. International Journal of Science and Research (IJSR). 2017;p. 459–461.
- [30] Rithika H, Santhoshi BN. Image text to speech conversion in the desired language by translating with Raspberry Pi. In: 2016 IEEE International Conference on

- [31] Sachin B, Rohan T, Harshranga P, Bhurchandi K. Substitute eyes for Blind using Android. In: India Educators' Conference (TIIEC), Texas Instruments; 2013. p. 38–43.
- [32] Sheth R, Rajandekar S, Laddha S, Chaudhari R. Smart white cane—an elegant and economic walking aid. American Journal of Engineering Research. 2014;3(10):84–89.
- [33] Salahat E, Qasaimeh M. Recent advances in features extraction and description algorithms: A comprehensive survey. In: 2017 IEEE international conference on industrial technology (ICIT). IEEE; 2017. p. 1059–1063.
- [34] Zahari M, Hamzah RA, Manap NA, Ibrahim H. Stereo Matching Algorithm Based on Census Transform and Segment Tree Cost Aggregation. In: Dang NHT, Zhang YD, Tavares JMRS, Chen BH, editors. Artificial Intelligence in Data and Big Data Processing. Cham: Springer International Publishing; 2022. p. 495–505.
- [35] Jabnoun H, Benzarti F, Amiri H. Object recognition for blind people based on features extraction. In: International Image Processing, Applications and Systems Conference; 2014. p. 1–6.
- [36] Mohane V, Gode C. Object recognition for blind people using portable camera. In: 2016 World Conference on Futuristic Trends in Research and Innovation for Social Welfare (Startup Conclave). IEEE; 2016. p. 1–4.
- [37] Islam R, Islam MR, Talukder KH. An approach to extract text regions from scene image. In: 2016 International Conference on Computing, Analytics and Security Trends (CAST). IEEE; 2016. p. 138–143.
- [38] Boldini A, Garcia AL, Sorrentino M, Beheshti M, Ogedegbe O, Fang Y, et al. An Inconspicuous, Integrated Electronic Travel Aid for Visual Impairment. ASME Letters in Dynamic Systems and Control. 2021 03;1(4). 041004. <https://doi.org/10.1115/1.4050186>. [https://asmedigitalcollection.asme.org/lettersdynsys/article-pdf/1/4/041004/6659699/aldsc\\_1\\_4\\_041004.pdf](https://asmedigitalcollection.asme.org/lettersdynsys/article-pdf/1/4/041004/6659699/aldsc_1_4_041004.pdf).
- [39] Niknejad N, Ismail WB, Mardani A, Liao H, Ghani I. A comprehensive overview of smart wearables: The state of the art literature, recent advances, and future challenges. Engineering Applications of Artificial Intelligence. 2020;90:103529.
- [40] Sun W, Liu J, Zhang H. When smart wearables meet intelligent vehicles: Challenges and future directions. IEEE wireless communications. 2017;24(3):58–65.

- [41] Ren S, He K, Girshick R, Sun J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2015;6:39:1137–1149. <https://doi.org/10.48550/arxiv.1506.01497>.
- [42] Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC. MobileNetV2: Inverted Residuals and Linear Bottlenecks. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2018;12:p. 4510–4520. <https://doi.org/10.1109/CVPR.2018.00474>.
- [43] Wang CY, Bochkovskiy A, Liao HYM. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors; 2022. <https://doi.org/10.48550/arXiv.2207.02696>.
- [44] Zhu X, Lyu S, Wang X, Zhao Q. TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-captured Scenarios. *Proceedings of the IEEE International Conference on Computer Vision*. 2021;8:2021-October:2778–2788. <https://doi.org/10.1109/ICCVW54120.2021.00312>.
- [45] Yang F, Zhang X, Liu B. Video object tracking based on YOLOv7 and DeepSORT; 2022. <https://doi.org/10.48550/arXiv.2207.12202>. Available from: <https://arxiv.org/abs/2207.12202>.
- [46] Bewley A, Ge Z, Ott L, Ramos F, Upcroft B. Simple online and realtime tracking. In: 2016 IEEE International Conference on Image Processing (ICIP); 2016. p. 3464–3468.
- [47] Sahbani B, Adiprawita W. Kalman filter and Iterative-Hungarian Algorithm implementation for low complexity point tracking as part of fast multiple object tracking system. In: 2016 6th International Conference on System Engineering and Technology (ICSET); 2016. p. 109–115.
- [48] Moryossef A, Tsochantaridis I, Dinn J, Camgöz NC, Bowden R, Jiang T, et al. Evaluating the Immediate Applicability of Pose Estimation for Sign Language Recognition. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW); 2021. p. 3429–3435.
- [49] Zhang F, Bazarevsky V, Vakunov A, Tkachenka A, Sung G, Chang CL, et al. MediaPipe Hands: On-device Real-time Hand Tracking. arXiv preprint arXiv:200610214. 2020;<https://doi.org/10.48550/ARXIV.2006.10214>.
- [50] Grishchenko I, Ablavatski A, Kartynnik Y, Raveendran K, Grundmann M. Attention Mesh: High-fidelity Face Mesh Prediction in Real-time; 2020. <https://doi.org/10.48550/arXiv.2006.10962>. Available from: <https://arxiv.org/abs/2006.10962>.

- [51] Wu CJ, Brooks D, Chen K, Chen D, Choudhury S, Dukhan M, et al. Machine learning at facebook: Understanding inference at the edge. In: 2019 IEEE international symposium on high performance computer architecture (HPCA). IEEE; 2019. p. 331–344.
- [52] Budrionis A, Plikynas D, Daniušis P, Indruļionis A. Smartphone-based computer vision travelling aids for blind and visually impaired individuals: A systematic review. *Assistive Technology*. 2022;34(2):178–194. PMID: 32207640. <https://doi.org/10.1080/10400435.2020.1743381>. <https://doi.org/10.1080/10400435.2020.1743381>.
- [53] Shrestha S, Lu D, Tian H, Cao Q, Liu J, Rizzo JR, et al. Active Crowd Analysis for Pandemic Risk Mitigation for Blind or Visually Impaired Persons. In: Bartoli A, Fusiello A, editors. Computer Vision – ECCV 2020 Workshops. Cham: Springer International Publishing; 2020. p. 422–439.
- [54] Mahendran JK, Barry DT, Nivedha AK, Bhandarkar SM. Computer Vision-based Assistance System for the Visually Impaired Using Mobile Edge Artificial Intelligence. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW); 2021. p. 2418–2427. Available from: [https://openaccess.thecvf.com/content/CVPR2021W/MAI/papers/Mahendran\\_Computer\\_Vision-Based\\_Assistance\\_System\\_for\\_the\\_Visually\\_Impaired\\_Using\\_Mobile\\_CVPRW\\_2021\\_paper.pdf](https://openaccess.thecvf.com/content/CVPR2021W/MAI/papers/Mahendran_Computer_Vision-Based_Assistance_System_for_the_Visually_Impaired_Using_Mobile_CVPRW_2021_paper.pdf).
- [55] Yuan Z, Azzino T, Hao Y, Lyu Y, Pei H, Boldini A, et al. Network-Aware 5G Edge Computing for Object Detection: Augmenting Wearables to 'See' More, Farther and Faster. *IEEE Access*. 2022;10:29612–29632. <https://doi.org/10.1109/ACCESS.2022.3157876>.
- [56] Gui W, Li B, Yuan S, Rizzo JR, Sharma L, Feng C, et al. An assistive low-vision platform that augments spatial cognition through proprioceptive guidance: Point-to-Tell-and-Touch. *IEEE International Conference on Intelligent Robots and Systems*. 2019 11;p. 3817–3822. <https://doi.org/10.1109/IROS40897.2019.8967647>.
- [57] Liu XJ, Fang Y. Virtual Touch: Computer Vision Augmented Touch-Free Scene Exploration for the Blind or Visually Impaired. In: 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW); 2021. p. 1708–1717.
- [58] Raheja J, Das K, Chaudhary A. Fingertip Detection: A Fast Method with Natural Hand. *International Journal of Embedded Systems and Computer Engineering (IJESCS)*. 2012 12;3.
- [59] Sural S, Qian G, Pramanik S. Segmentation and histogram generation using the HSV color space for image retrieval. In: Proceedings. International Conference on Image Processing. vol. 2; 2002. p. II–II.

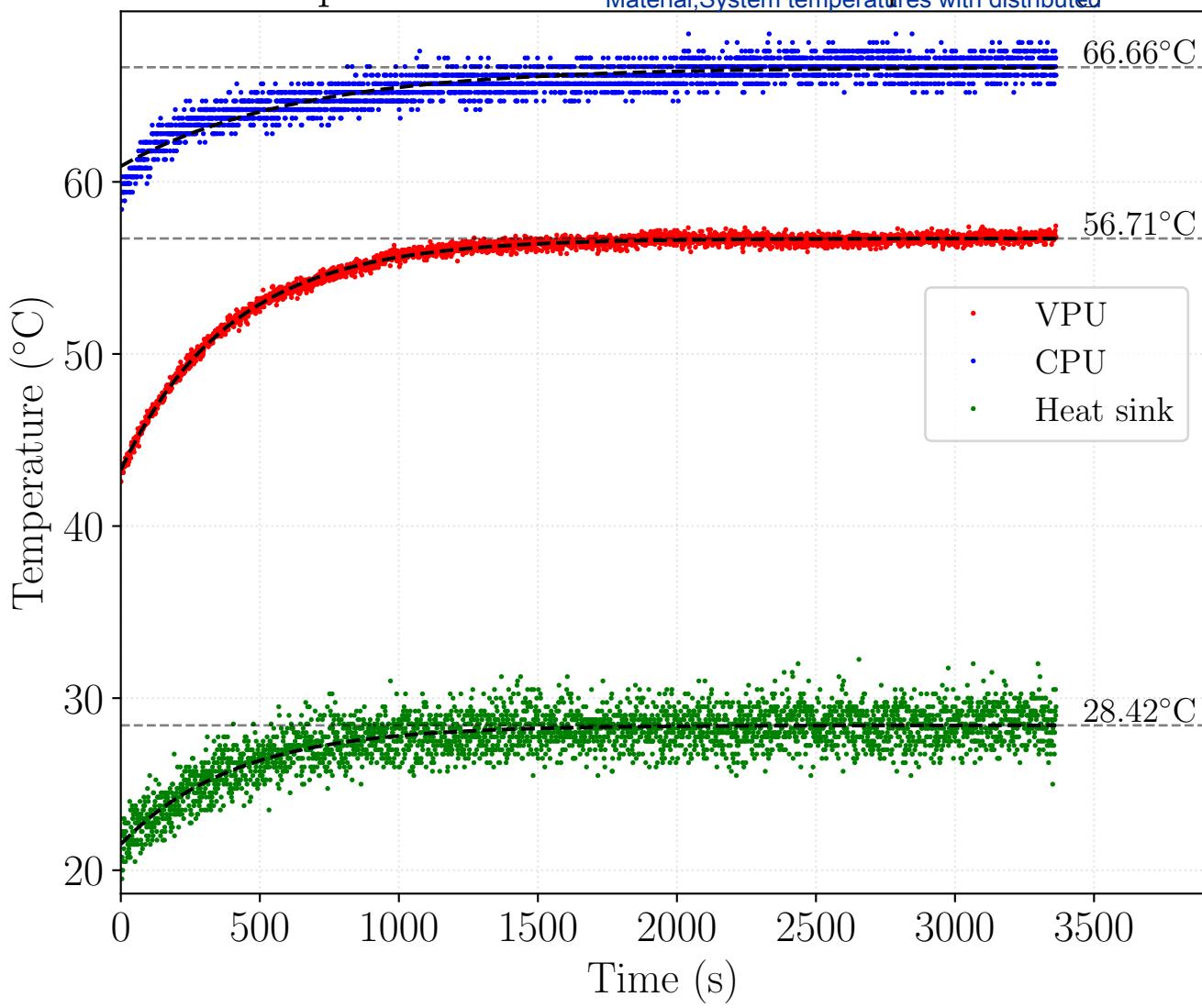
- [60] Liu H, Liu R, Yang K, Zhang J, Peng K, Stiefelhagen R. HIDA: Towards Holistic Indoor Understanding for the Visually Impaired via Semantic Instance Segmentation with a Wearable Solid-State LiDAR Sensor. CoRR. 2021;abs/2107.03180. [2107.03180](https://arxiv.org/abs/2107.03180).
- [61] Kuribayashi M, Kayukawa S, Vongkulbhaisal J, Asakawa C, Sato D, Takagi H, et al. Corridor-Walker: Mobile Indoor Walking Assistance for Blind People to Avoid Obstacles and Recognize Intersections. Proc ACM Hum-Comput Interact. 2022 sep;6(MHCI). <https://doi.org/10.1145/3546714>.
- [62] Jafri R, Ali SA, Arabnia HR, Fatima S. Computer vision-based object recognition for the visually impaired in an indoors environment: a survey. VISUAL COMPUTER. 2014 NOV;30(11):1197–1222. <https://doi.org/10.1007/s00371-013-0886-1>.
- [63] Plikynas D, Žvironas A, Budrionis A, Gudauskis M. Indoor Navigation Systems for Visually Impaired Persons: Mapping the Features of Existing Technologies to User Needs. Sensors. 2020;20(3). <https://doi.org/10.3390/s20030636>.
- [64] Havik EM, Kooijman AC, Steyvers FJJM. The Effectiveness of Verbal Information Provided by Electronic Travel Aids for Visually Impaired Persons. journal OF VISUAL IMPAIRMENT & BLINDNESS. 2011 OCT-NOV;105(10):624–637. <https://doi.org/10.1177/0145482X1110501009>.
- [65] Giudice NA, Whalen WE, Riehle TH, Anderson SM, Doore SA. Evaluation of an Accessible, Real-Time, and Infrastructure-Free Indoor Navigation System by Users Who Are Blind in the Mall of America. journal OF VISUAL IMPAIRMENT & BLINDNESS. 2019 MAR-APR;113(2):140–155. <https://doi.org/10.1177/0145482X19840918>.
- [66] Kuriakose B, Shrestha R, Sandnes FE. Multimodal Navigation Systems for Users with Visual Impairments—A Review and Analysis. Multimodal Technologies and Interaction. 2020;4(4):73.
- [67] Berhe MK. Ergonomic Temperature Limits for Handheld Electronic Devices. 2007 Proceedings of the ASME InterPack Conference, IPACK 2007. 2010 1;2:1041–1047. <https://doi.org/10.1115/IPACK2007-33873>.
- [68] Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, et al. Microsoft COCO: Common Objects in Context. In: Fleet D, Pajdla T, Schiele B, Tuytelaars T, editors. Computer Vision – ECCV 2014. Cham: Springer International Publishing; 2014. p. 740–755.
- [69] Padilla R, Passos WL, Dias TLB, Netto SL, da Silva EAB. A Comparative Analysis of Object Detection Metrics with a Companion Open-Source Toolkit. Electronics. 2021;10(3):279. <https://doi.org/10.3390/electronics10030279>.

- 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  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65
- [70] Ibáñez N.: BVI Sistem: Development of an assisted vision system for blind people based on hand movement and artificial intelligence. GitHub. Available from: <https://doi.org/10.5281/zenodo.7569573>.



# Temperatures with distributed processing

Click here to access/download Supplementary Material, System temperatures with distributed



# Temperatures without distributed processing

Click here to access/download Supplementary Material, System temperatures without distributed processing

