Computer Vision-based Assistance System for the Visually Impaired Using Mobile Edge Artificial Intelligence

Jagadish K. Mahendran[1] Daniel T. Barry[2] Anita K. Nivedha[3] Suchendra M. Bhandarkar[4]

[1]Kutir Technologies Corporation, Vancouver, BC, Canada [2]Denbar Robotics, Boston, MA, USA
[3]Molecular Forecaster Inc., Montreal, QC, Canada [4]Department of Computer Science, University of Georgia, Athens, GA, USA
jagadishkmahendran@gmail.com dbarry@denbarrobotics.com anita.nivedha@molecularforecaster.com suchi@uga.edu

Abstract

Despite significant recent developments, visual assistance systems are still severely constrained by sensor capabilities, form factor, battery power consumption, computational resources and the use of traditional computer vision algorithms. Current visual assistance systems cannot adequately perform complex computer vision tasks that entail deep learning. We present the design and implementation of a novel visual assistance system that employs deep learning and point cloud processing to perform advanced perception tasks on a cost-effective, low-power mobile computing platform. The proposed system design circumvents the need for expensive, power-intensive Graphical Processing Unit (GPU)-based hardware required by most deep learning algorithms for real-time inference by employing instead edge Artificial Intelligence (AI) accelerators such as the Neural Compute Stick-2 (NCS2), model optimization techniques such as OpenVINO, and TensorFlow Lite, and smart depth sensors such as OpenCV AI Kit-Depth (OAK-D). Critical system design challenges such as training data collection, real-time capability, computational efficiency, power consumption, portability and reliability are addressed. The proposed system includes more advanced functionality than existing systems such as assessment of traffic conditions and detection and localization of hanging obstacles, crosswalks, moving obstacles and sudden elevation changes. The proposed system design incorporates an AI-based voice interface that allows for user-friendly interaction and control and is shown to realize a simple, cost-effective, power-efficient, portable and unobtrusive visual assistance device.

Keywords: Edge Artificial Intelligence (AI), Mobile Computing, Visual Assistance Device, Deep Learning

1. Introduction

Visual impairment has been a global issue for several decades. According to a report published by the World Health Organization titled "*Global Data on Visual Impairments 2010",* of an estimated 285 million visually impaired people globally in 2010, 246 million suffered from low levels of vision and 39 million were completely blind [1]. It was estimated that 65% of the visually impaired and 82% of the blind were 50 years of age or older. The major causes for visual impairment were noted to be uncorrected refractive errors (43%) and presence of cataract (33%) [1]. The causes for blindness included cataract (51%), glaucoma (8%), age-related macular degeneration (5%), childhood blindness and corneal opacities (4%), and undetermined causes (21%) [1]. In the years between 1990 and 2015 there has been considerable improvement globally in the relative percentage of people with visual impairment (from 4.58% to 3.38%), considering a 38% increase in the overall world population and near doubling of the population of adults 50 years and older [2], [3], [4]. Increased public awareness, affordable eye health care services and decline in poverty levels are among the key factors for this positive development. However, there has been a significant increase in the number of visually impaired people on account of the steadily increasing population, especially the older population [2], [3], [4]. By 2030, there are estimated to be 385 million visually impaired people globally, of which 330 million will suffer from low levels of vision and 55 million will be legally blind [2].

Common challenges faced by visually impaired people include dependency on others, unemployment, reduced social interactions, difficulty reading, writing, performing daily activities, transportation, medication handling, and operating devices such as phones and laptops, anxiety in crowded areas, ambulatory injuries and victimization by seemingly overly helpful individuals [5], [6], [7], [8], [9]. Specifically, while navigating outdoors visually impaired people are unable to accurately assess the traffic, sidewalk and road conditions. Guide dogs, walking canes and Global Positioning System (GPS)-enabled devices are commonly employed to deal with these situations. While guide dogs can detect obstacles, their communication methods with human dependents are often unclear. Walking canes are extremely effective in spotting ground-level anomalies; however, their use entails constant probing, ineffective in detecting overhanging obstacles like tree branches, open windows or wires. GPS-enabled devices can help with routing but not with obstacle detection. An advanced Artificial Intelligence (AI)-based perception system can be deemed to be the best means of assisting the visually impaired by providing a comprehensive, rich understanding of the environment and enabling safe navigation.

However, developing an advanced AI-based perception system for visual assistance is far from trivial. Accurate

modelling of outdoor environments entails comprehensive training of deep learning models, which is a highly data-intensive process that requires powerful Graphical Processing Unit (GPU)-based hardware for real-time inference. High-performance GPUs also contribute to an unreasonable form factor, high battery power consumption and high cost resulting in a physical setup that is heavy, expensive, obtrusive and not user-friendly.

In this paper, we propose a computer vision-based visual assistance system to overcome these limitations using edge AI accelerator devices such as the Intel's *Neural Compute Stick-2* (NCS2) in conjunction with model conversion and optimization techniques such as quantization using *OpenVINO* and *TensorFlow Lite*. We also employ a state-of-the-art *OpenCV AI Kit-Depth* (OAK-D) sensor that provides RGB images along with depth information using stereo vision. More importantly, the OAK-D sensor has an inbuilt on-chip AI processor, i.e., the *Intel MyriadX*, capable of running inference models on the captured video data *before* transmitting the video frames to the host machine. We also leverage existing pretrained *Advanced Driver Assistance System* (ADAS) models used in autonomous vehicles to perform complex perception tasks such as the detection of roads, sky, crosswalk, curbs, cars and vegetation amongst other common object classes. As a result, we have developed a visual assistance system with a simple form factor that is cost effective, portable, and almost unnoticeable as an assistive device. The proposed system design shows how deep learning algorithms can be efficiently incorporated within computer vision-based visual assistance systems. More importantly, through this project we hope to contribute to the quality of life of visually impaired people by increasing their involvement in and enjoyment of daily activities.

The remainder of this paper is organized as follows: we discuss related work and similar projects in Section 2; in Section 3 we detail the design of the proposed visual assistance system and the associated deep learning approaches; in Section 4, we describe the hardware and physical setup of the proposed system and in Section 5, we present the performance evaluation results of the proposed system. Finally, we conclude the paper in Section 6 and present an outline for future enhancements and extensions to this project.

2. Related Work

Visual assistance devices are commonly classified as *Electronic Travel Aids* (ETAs) that provide information about the environment through a convenient user interface. Various approaches have been proposed in the literature for the design of visual assistance systems based on the underlying sensory systems, hardware configuration, physical setup, data inference techniques and user interface. The most used sensor types include ultrasound, sonar, laser, RGB CCD camera, infrared (IR) camera and GPS. Some approaches convert the input sensor data to other modalities [11], [12]. For the user interface, audio transmitted via earphones or hand gloves equipped with buzzers or tiny vibrating actuators are typically used.

Early visual assistance system designs were based on projecting a camera image onto the human skin using vibrating motors [10] and sensor modality conversion, where ultrasonic waves were converted to the audible range and the converted audio was used to understand the environment [11], [12]. Although the *vOIce* system [12], where visual image data are converted to human audible frequency, showed promising results these systems were typically slow, physically uncomfortable, and obtrusive, provided only very coarse information about the surroundings and required extensive user training to be used effectively. Early visual assistance systems based on GPS data [13], focused primarily on navigation (i.e., neither collision avoidance nor obstacle detection was performed) and often suffered from signal loss, especially in indoor environments and urban areas. Visual assistance systems based on RFID technology provided good localization in indoor environments wherein RFID tags were physically placed [14]. Since RFID sensing methods provide a range rather than an accurate geolocation of the tags, the resulting localization errors were unacceptable in certain situations.

In the past few decades, as the sensor and computing technologies have evolved remarkably, so have visual assistance systems. *NavBelt* [15] is a real-time visual assistance system that uses eight ultrasonic sensors mounted on a waist belt worn by the user with a computing unit located in a backpack. *GuideCane* [16] is a wheel-based cane with an ultrasonic sensor attached to a main processing unit that is mounted on the cane. Bousbia-Salah et al. [17] describe a visual assistance system that uses two ultrasonic sensors strapped onto the user's shoulders accompanied by an accelerometer, with a foot switch for error control. The *CyARM* system [18] consists of a handheld device with two ultrasonic sensors to detect obstacles, coupled with a wire-enabled user interface mounted through a waist belt. While ultrasonic sensors are low-cost and fast, they fail to provide accurate geometric descriptions of the obstacles encountered and are prone to errors caused by noise and signal reflections. Extensive reviews of ETAs and the challenges faced in their design and deployment can be found in [19] and [20].

Recently, promising advancements in the design of ETAs have been made using computer vision-based approaches. Tapu et al. [21] use a smartphone camera mounted on a chest harness to detect moving obstacles using the multiscale Lucas-Kanade feature tracking algorithm. Object detection is performed by classifying a *Bag of Visual Words* (BoVW) and *Histogram of Oriented Gradients* (HOG) features using linear classifiers such as the *Support Vector Machine* (SVM) and image ranking

methods. Jabnoun et al. [22] propose an ETA that uses SIFT feature-based object detection on video streams. These ETAs employ traditional computer vision methods for object detection that are not robust in real-world environments. Moreover, they lack 3D information, in that the user does not know how far away the obstacles are.

The *Electron-Neural Vision System* (ENVS) uses eyewear-based stereo cameras to obtain 3D descriptions of the environment from depth images [23]. The system is also equipped with a GPS, portable computer, *Transcranial Electrical Nervous Stimulation* (TENS) unit and a TENS-based glove as the user interface. Rodriguez et al. [24] propose a stereo vision-based system where plane segmentation is performed to detect ground pixels and a polar grid notation used to detect obstacles within depth images. The stereo cameras are mounted in the chest region, coupled with an audio-based user interface. Johnson and Higgins [25] proposed a stereo vision-based scheme, with stereo cameras mounted in the hip region on a tactile belt with 14 vibrating motors worn by the user, that also serves as a user interface. While the above systems provide advanced, accurate 3D perception of the environment and have robust, reliable obstacle detection capabilities by exploiting stereoscopic vision, they lack the scene understanding capabilities needed for assessment of traffic, road and sidewalk conditions, and advanced deep learning-based perception capabilities, such as image classification, objection detection and semantic image segmentation.

There has been significant recent progress in the design and development of visual assistance systems that derive a richer understanding of the user's environment. State-of-the-art visual assistance systems are designed to use a variety of sensor types including conventional CCD RGB cameras, stereoscopic cameras, GPS-enabled devices, and ultrasound and RFID sensors. Several systems incorporate sensor modality conversion techniques in conjunction with an intuitive user-friendly interface. Existing systems that use camera-based sensors typically employ traditional computer vision algorithms on predefined image features such as SIFT, HOG and BoVW, to mention a few, that are observed to perform adequately in constrained, well-defined environments but fail to generalize to a real-world settings characterized by greater variability. Also, existing camera-based systems do not adequately tackle advanced perception tasks such as assessment of traffic conditions and elevation changes (e.g., curb detection), understanding of traffic signs, detection of crosswalks, etc. Several conventional camera-based systems have a bulky, obtrusive physical setup, which is not user-friendly and may draw unwarranted attention in public spaces. Consequently, there is a need for visual assistance systems that execute deep learning-based inference algorithms for advanced perception tasks, while preserving a simple, unobtrusive form factor and consuming sufficiently low battery power. This would ensure user mobility and avoid unwarranted attention during user ambulation in public spaces.

In this paper we propose a novel and practical design for a visual assistance system using a smart stereo vision sensor called OAK-D and edge AI devices that can overcome the limitations of existing approaches by using edge AI technology. We propose a visual assistance system with deep learning capabilities to perform advanced computer vision tasks, such as object detection and semantic image segmentation in real time using edge AI devices such as the Intel's NCS2. The proposed system is designed to perform sophisticated scene understanding tasks such as detection of roads, curb entry/exit locations, crosswalks, assessment of traffic conditions by detecting traffic lights and congestion, reading of traffic signs and street names using state-of-the-art computer vision and deep learning methods. The proposed system can perform 3D point cloud processing to detect elevation changes at curb entry/exit locations.

The proposed system is equipped with a GPS-enabled device for geolocalization that can save custom locations for convenience. The GPS coordinates along with a snapshot of the current location can be shared with preferred contacts over a short message service (SMS) if the user needs emergency assistance. A user-friendly, customizable, AI-based interactive voice interface equipped with speech recognition provides the user periodic updates about the environment. The physical setup consists of a chest-mounted OAK-D sensor placed inside a vest and connected to a computing unit placed in a small backpack. Wireless Bluetooth earphones with a microphone are used for voice-based interactions. The setup is unobtrusive and not noticeable as an assistive system. Interviews were conducted with visually impaired people, including those from non-profit organizations such as *LightHouse for the Blind Inc*., to comprehend and catalog the common daily challenges they face. We prioritized and ranked the identified challenges based on these interviews and addressed them in our design.

3. System Design and Methods

The AI software system is divided primarily into the perception module and user interface as depicted in Fig. 1. The perception module is further subdivided into three major submodules, i.e., the primitive perception, advanced perception and localizer submodules. The primitive perception submodule deals primarily with obstacle presence detection using depth information while the advanced perception submodule provides the user with a more comprehensive description of the environment using complex computer vision algorithms for object detection, semantic image segmentation and image classification. The localizer submodule is used to geolocate the user within the environment. The user interface module includes the speech recognition, text-to-speech and SMS submodules.

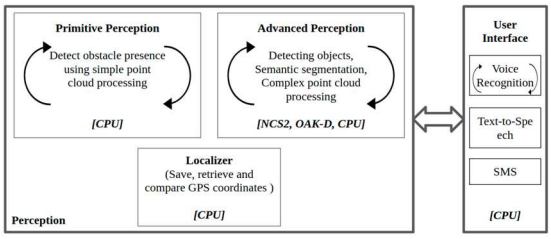
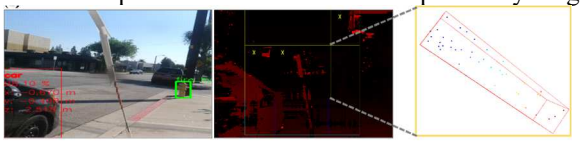


Figure 1: System Design Block Diagram. Each block contains its task description and hardware requirements. Blocks with cyclic arrows are executed in a continuous loop, others are triggered by external events like voice control.

3.1 Primitive perception

In the primitive perception submodule, obstacles within a certain range are detected and the user updated sufficiently in advance to avoid collisions. It is designed to be simple and fast, as the priority is to detect obstacles rather than understand them using depth data from the OAK-D sensor. The raw depth frame is divided into multiple grids and grid positions are chosen to cover various locations around the user (right, left, center, top) to handle various scenarios including hanging obstacles. The grid depth data is converted to a point cloud format for further processing using the right camera's intrinsic matrix obtained via camera calibration. Point cloud processing is performed using the Open3D software library. In each grid, the number of points within a specified distance (1.5m), after noise removal, is computed and if this count exceeds a threshold (800 points found to be optimal) the grid is marked to contain an obstacle. A cuboid is fitted to the grid points to estimate the rough shape of the obstacle. The depth data from the OAK-D sensor is not reliable for closer ranges (< 0.8m), so the proximity range cannot be reduced further.

This submodule allows for robust obstacle detection for many shapes at various heights including wires, kitchen slabs etc. at a high inference rate on the host CPU machine. A few sample outputs along with the RGB image and fitted cuboids are shown (Figs. 2, 3). The grid positions marked with '**X**' represent obstacles within the proximity range.



(a) (b) (c)

Figure 2: Sensor recognizing a wire from a utility pole. **(a)** color image **(b)** center grid with detected obstacle **(c)** cuboid fitted to point cloud in grid

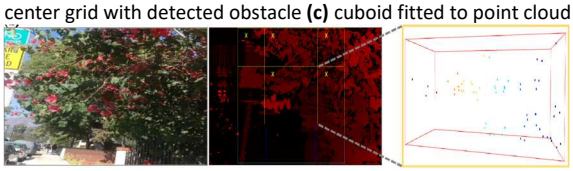


Figure 3: Sensor recognizing tree branches extending onto sidewalk.

3.2 Advanced perception

The advanced perception submodule performs sophisticated scene understanding tasks such as image classification, object detection, semantic image segmentation and 3D point cloud processing to extract complex perceptual information about the environment. We treat the visual assistance system design problem as we would the design of self-driving vehicles, requiring reliable sensors, large training datasets, the training of complex deep learning models with high accuracy and fast inference rates. In our design, we leverage techniques and models developed in the domain of self-driving vehicles, however, we are constrained by cost, computational complexity, battery power consumption and physical form factor. We propose a solution using edge AI optimization techniques that allow us to implement computationally intensive deep learning models on small and cheap edge devices. These models are further optimized using OpenVINO optimization techniques, permitting the execution of multiple objection detection, image classification and semantic image segmentation models in real time to provide accurate visual assistance. Unlike existing systems, our solution can detect people, cars, traffic lights, yellow pavements that aid the blind, traffic signs such as stop signs, and signs denoting sidewalk closures, pedestrian crossings and speed limits. Our semantic image segmentation models can detect the road, curb and road marking pixels.

A large collection of datasets was used for training our model, including the Google Open Image (GOI) dataset, Laboratory for Intelligent and Safe Automobiles (LISA) traffic signs dataset, German Traffic Sign Recognition Benchmark (GTSRB) dataset, Traffic Cone dataset and Cityscapes dataset. We also collected and labelled several thousand custom images to create our own dataset for each camera position by walking on the sidewalks in the downtown and nearby areas of Monrovia, CA at various times of the day. Models pretrained on the PASCAL VOC dataset from Luxonis' DepthAI library [26] were also used. As the GOI dataset is large, images with class labels relevant to our project, i.e., traffic lights, traffic signs and street names, were chosen and their labels converted to the PASCAL VOC format for training purposes. The GOI dataset had some labelling inconsistencies which were manually corrected. The custom dataset for detecting object classes such as traffic lights, traffic signs, fire hydrants etc. includes a total of 599 images, 10% of which are from the GOI dataset and other sources. For traffic signs classification we use a combination of the LISA and GTRSB datasets and our custom dataset which includes ~560 images. Fig. 4 shows some sample images collected for our custom dataset available at: https://drive.google.com/drive/folders/1HgLOO-HA3YntmjhF0FZvdrJhK-FmV-2C?usp=sharing.

Figure 4: Sample images from our custom dataset

Common object classes observed while walking on sidewalks include pedestrians, dogs, cats, bikes, trees, fences, road, cars, motorbikes, bicycles, traffic signs such as stop signs and signs denoting sidewalk closures, pedestrian crossings and speed limits, traffic lights, traffic cones, street names, public trash cans, sign boards etc. While some of these classes can be detected using object detection models, others require semantic image segmentation models. We trained custom object detection models and also used existing pretrained models from the OpenVINO and TensorFlow model zoos which include DepthAI's SSD PASCAL object detection model, OpenVINO's ADAS models and TensorFlow Lite's model pretrained on the Cityscapes and ADEK20 datasets.

3.2.1 Detecting Objects

Apart from DepthAI's SSD-MobileNet object detection model pretrained on the PASCAL VOC dataset, custom models were trained to detect traffic-related classes such as traffic signs, traffic lights, traffic cones, fire hydrants, yellow pavements, crosswalk buttons, public trash cans etc. The SSD-MobileNet model [27] was chosen for its compactness, speed and single-stage detection capability. The model was trained on multiple image resolutions ranging from 600×400 pixels to 300×300 pixels for 300,000 steps with an initial learning rate of 0.0008, decay rate of 0.95 and batch size of 24. A training : validation : test data split of 70:20:10 was used. Best results were obtained on an image resolution of 300×300 pixels with a mean average precision (mAP) of 0.62.

Lightweight models like SSD-MobileNet suffer from lack of accuracy resulting in false negatives (FN) (i.e., missed detections) and false positives (FP). To reduce the FN rate the model threshold was reduced to 0.3, which further increased the FP rate mostly for background classes such as leaves, sky and distant buildings. The FPs are largely suppressed by training another lightweight CNN-based image classifier with a CONV=>RELU=>BN=>POOL layer and 2 sets of (CONV=>RELU=>CONV=>RELU)*2=>POOL layers as in the *TrafficSignNet* model (Fig. 5) [28]. The FPs were further removed by maintaining detection counts for > 2 seconds, with detections that occurred for less than a threshold count deemed weak detections, and ignored. The object detection pipeline block diagram is shown in Fig. 6.

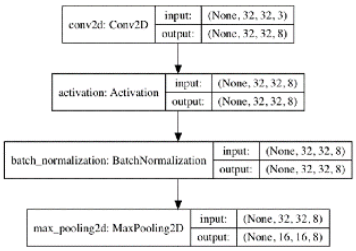


Figure 5: The *TrafficSignNet* submodule

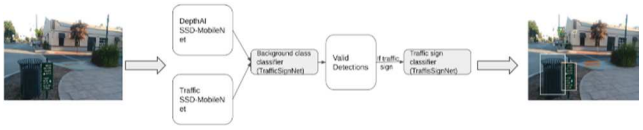


Figure 6: Object detection pipeline

Figure 7: Object detection model detecting traffic signs, traffic lights, road signs, trash can and yellow pavement.

Table 1: Quality metrics of road segmentation model on the *Mighty AI* dataset (table obtained from openvinotoolkit.org)

Label	IOU	ACC
mean	**84.4%**	**90.1%**
BG	98.6%	99.4%
Road	95.4%	97.4%
Curbs	72.7%	83.1%
Marks	70.8%	80.6%

IOU=Intersection Over Union; **ACC**=Accuracy

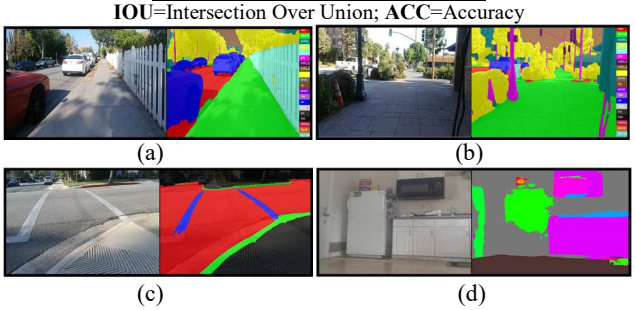


Figure 8: (a) Cityscapes semantic image segmentation model OpenVINO **(b)** TFLite ADE20K model **(c)** Road segmentation model OpenVINO **(d)** TFLite ADE20K Indoor segmentation model.

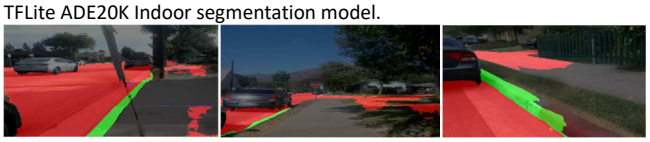


Figure 9: Road segmentation model failure cases on sidewalk.

3.2.2 Semantic scene understanding

Object detection by itself is not adequate for complete scene understanding, especially in the case of recognition of sidewalks, roads, crosswalks and vegetation. As with self-driving vehicles, semantic image segmentation models are used for this purpose. Heavyweight models such as the UNet and PSPNet are not suitable for this application as they require powerful GPUs for real-time inference. The popular OpenVINO's ADAS pretrained semantic image

segmentation models such as *semantic-segmentation-adas-0001* [29] and *road-semantic-segmentation-adas-0001* [30] along with TensorFlow Lite's DeepLabv3 MobileNet models pretrained on the Cityscapes and ADE20K datasets were used instead. The *semantic-segmentation-adas-0001* model produced the best results for outdoor scenarios. However, this model (FP16) is constrained by its size and cannot be loaded onto either OAK-D or NCS2 but ran at a speed of ~2.3 fps on the host CPU. *The road-segmentation-adas-0001* model is relatively smaller and executable on a multi-Myriad platform (comprising of OAK-D and NCS2), and is trained on 4 classes: road, sidewalk, road markings and background with model performance metrics shown in Table 1. This model is also used to detect crosswalks at intersections. The TensorFlow Lite models were slower (~0.4 fps); however, the pretrained ADE20K model yielded the best results for indoor scenarios. Also, since the pretrained models are often trained on datasets collected in a different setting, i.e., while driving with a camera mounted on the car, these models sometimes fail to differentiate between the road and sidewalk (Fig. 9). Fig. 8 shows sample prediction outputs from various types of semantic image segmentation models.

3.2.3 Crosswalk detection

A common approach for road lane detection is to perform edge detection followed by the Hough transform to detect lanes. While this technique generally works, it requires the camera to be mounted on the car providing a smooth and contrasting image of the road with lanes. Also, only a predetermined subregion of the image is processed for lane detection in these methods. In this project, since the camera is at varying distances from the road, the road texture is enhanced, producing roughly textured and noisy images which overwhelm conventional image processing routines such as the Canny edge detector and color detector. Since we operate in an outdoor setting, most color detection methods are unreliable and are observed to perform poorly in bright sunlight and in the presence of reflections.

We performed road marks segmentation using the *road-segmentation-adas-0001* model which provided better results for our application than traditional methods as shown in Fig. 10. The model predictions are processed further for noise removal followed by contour analysis. The size and shape of each connected component in the image is characterized in terms of its area, convex hull, orientation and solidity. The closed contours or blobs that are tall, with solidity values > 0.8, and orientation angles between 35° – 65° and 100° – 140° are chosen to ensure that the user is facing the crosswalk line. The sample outputs of the road marks segmentation model are shown in Fig. 10. In cases where the lines are faded or covered with tire marks, the model performance suffers. The chest mounted camera provided better results than the hip mounted one probably due to the road textures being more prominent in the latter case. Currently, the crosswalk detection is performed only within a certain spatiotemporal range after detecting stop signs to reduce computational complexity.

(a) (b) (c) (d)

Figure 10: (a) Segmentation results **(b)** Connected component extraction for road markings **(c)** Noise removal and dilation **(d)** Contour analysis and final detection.

3.2.4 Elevation change detection

Failure to detect elevation changes such as when entering/exiting a curb, climbing/descending staircases can result in serious injuries. 3D point cloud data is typically used to estimate elevation. Presently, we are unable to estimate the surface normals reliably using OAK-D depth data due to its low depth resolution (96 depth levels). Consequently, we could not employ traditional point cloud processing methods for modelling elevation changes. Alternate approaches such as machine learning on depth images and semantic image segmentation were employed.

Common point cloud-based machine learning (ML) models such as PointNet require a fixed number of features. While a mesh can be sampled to generate a fixed number of points, point clouds must be first converted to a mesh which is a computationally intensive procedure. Instead, we use an ensemble of 2 MiniVGGNet classifiers [31] for color images and depth images that are trained for up-curb, down-curb and flat-surface classes. MiniVGGNet is a miniature version of VGGNet that is small, fast and can be used for simpler classification problems. A separate dataset comprising of ~9000 RGB color and depth images was collected in the vicinity of curb areas for this purpose. For a training:validation data split of 75:25 on the depth image dataset, learning rate of 0.001, batch size of 32 and 50 epochs, the accuracy obtained was 96%. Similarly, for the RGB dataset with a learning rate of 0.001, batch size of 12 and 50 epochs, the accuracy obtained was 97%. The entire dataset (with sample images shown in Fig. 11) is shared at https://drive.google.com/drive/folders/1ZpbacfzLHytdh07SoEwzpHpGz0iblJp0?usp=sharing

Figure 11: Curb dataset for elevation detection

The MiniVGGNet models were initially trained using TensorFlow. On converting them to OpenVINO format we observe ~13× boost in inference speed with negligible decrease in accuracy. While this ML-based approach performs reasonably well, using a traditional point cloud-based approach may potentially provide more robust results, and is planned in our future work. The current implementation can also be extended to staircase detection.

In detecting elevation changes, we observe that moving the camera to the waist region produced results comparable to those from the chest mounted camera. This suggests that elevation change detection could benefit from an additional OAK-D camera. Note that an additional camera should not affect the system performance as the MiniVGGNet is a lightweight model and can be run directly on the OAK-D sensor.

3.2.5 Localization

A VK-162 G-Mouse USB-enabled GPS is used for geolocalization. The user can save common locations such as a friend's apartment, grocery store, gym etc. tagged with their preferred names. For the purpose of localization, upon the user's request, the system can read out the saved locations within a certain distance. Additionally, the GPS coordinates can be shared with known contacts via text-based SMS service using the Twilio interface. Since an SMS service requires an internet connection, we used cellular tethering for the internet in our experiments. This feature can also capture images to be shared along with the GPS coordinates in the SMS text.

3.3 System User Interface

Significant attention was paid to providing a comfortable, friendly user experience by avoiding continuous bombardment of the user with information - a common issue with most existing apps. The complete system can be controlled using a voice-based interface. To reduce the annoyance of trivial messages, most updates are provided only upon user request with exceptions of critical updates related to user safety. Primitive perception updates that indicate a possible collision with an obstacle along with significant elevation changes are considered critical.

Bluetooth-enabled wireless earbuds were used in this project. Common text-to-speech software packages such as GoogleTTS, Microsoft Speech Engine and TTS-Watson require an internet connection. Among the offline packages, Festival was observed to perform better than Pyttsx3. For speech recognition, OpenVINO's pretrained model, CMU's PocketSphinx and the Vosk framework were tested. While the trigger-based word recognition in PocketSphinx was relatively faster, it resulted in lower speech recognition accuracy than OpenVINO. However, the Vosk framework was ultimately used, due to higher speech recognition accuracy. For predefined words such as the object detection/segmentation classes Google's audio clips were downloaded and played using Python's Playsound package. Trigger words are used to trigger system features such as to start the system (trigger word: "start"), describing detected objects using analog clock-angles (3 to 9 o'clock) in the scene (trigger word: "describe"), saving landmarks (trigger word: "save location"), listing nearby saved locations (trigger word: "locate") and sending GPS location over SMS (trigger word: "send location"). These features are overridden if there are critical updates. On trigger word "describe", the detected objects in the scene are listed along with their location angles represented in clock notation between 3 to 9 o'clock traversed anticlockwise. Speech recognition and text-to-speech conversion generally result in slow and blocking synchronous function calls. This reduces system performance by lowering the inference speed and producing unpredictable behavior. Hence the user interface stack is made asynchronous and runs in parallel along with the perception and localizer stacks.

4. System Hardware Description

We propose a simple system that does not entail handheld devices such as a cane, laser sensor or a guide dog. The system comprises a small backpack to hold a small host computing unit (Raspberry Pi, Chromebook or laptop) and a battery power unit. An OAK-D sensor (Fig. 12 (a)) is mounted in the chest region inside a vest (Fig. 12 (b)). The sensor does not have to be at the same location every time as there is no calibration involved. The sensor is then connected to the computing unit in the backpack (Fig. 13). The OAK-D sensor is chosen as it contains a built-in AI processing unit allowing most of the computer vison tasks to be performed directly on the sensor chip. The inference data is collected by the host and updates provided to the user via a voice interface using Bluetooth-enabled wireless earphones. A USB-enabled GPS device connected to the host computing unit is mounted outside, over the backpack. The OAK-D sensor is powered by a compact 10000 mAh pocket-sized battery pack for up to ~8 hours. A five-year old Lenovo Yoga laptop (8GB RAM, Intel i7 processor) is used as the host computing unit along with a neural compute stick. This setup is convenient to wear, unobtrusive, not noticeable as an assistive device and in conformity with a majority of user requests resulting from our interviews.

5. Experimental Evaluation and Results

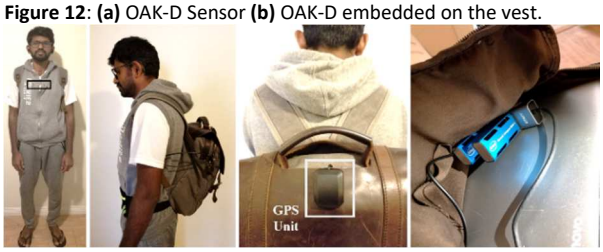
We targeted lightweight models such as the SSD MobileNet for object detection to ensure higher inference rates of ~30fps just on the OAK-D sensor. This allowed us to run multiple object detection models simultaneously on the CPU, NCS2 and OAK-D, to detect as many object classes as possible in real time. Fig. 14 depicts the real-time perception performance of our system near a crosswalk intersection. Fig. 13 depicts the system performance in the context of detection of traffic lights, cars, road markings, yellow pavements, bicycles etc. on Myrtle Avenue, in the downtown area of Monrovia, CA.

The traffic signs were classified using TrafficSign Net. The model performance is shown in Table 2. The model model was trained with a decaying learning rate of 0.0001, batch size of 64 and 100 epochs. Popular traffic datasets such as LISA and GTSRB were used along with our dataset. The detections were mapped to the depth image to compute their distances from the user. The TrafficSigNet model, after conversion to the OpenVINO format, results in

inference rates > 60 fps on the OAK-D sensor and ~100 fps on the CPU. The SSD-MobileNet model has an inference rate of ~30fps. Since OAK-D does not support multi-object detection models, the traffic object detection model was run on the external NCS2 devices. The overall system with primitive perception, object detection models and classification models has an inference rate of ~22fps.

(a) (b)

Figure 12: **(a)** OAK-D Sensor **(b)** OAK-D embedded on the vest.



(a) (b) (c) (d)

Figure 13: Physical setup showing **(a)** front view – sensors marked with black rectangular box **(b)** side view **(c)** GPS unit placed over the backpack and connected to the computing unit inside. **(d)** The computing unit inside the backpack along with NCS2s.

(a) (b) (c)

Figure 14: Real-time perception performance **(a)** object detection models detecting a stop sign, fire hydrant and car **(b)** road and curb pixels detection using road-segmentation-adas-0001 model **(c)** depth image processing for obstacle detection.

However, running semantic image segmentation models simultaneously along with other models is nontrivial. None of the semantic image segmentation models were able to run on a single NCS2. The road-segmentation-adas-0001 semantic image segmentation model is the fastest but requires a CPU or multiple NCS2 devices. The entire system, comprising this model, the primitive perception and object detection pipeline can infer at a satisfactory ~10 fps. However, other semantic image segmentation models suffer from high computational complexity and present a performance bottleneck. Models such as semantic-segmentation-adas-0001 and DeepLabV3 MobileNet can infer at 1.56 fps and 0.42 fps respectively while also requiring a host CPU. The performance of the whole system suffered upon using these models along with the other models. Currently, these heavyweight models are used only upon the user's request and the user is required to remain stationary for safety.

The project source code is available at https://github.com/jaggiK/cv_visual_assistance

(a) (b)

Figure 15: (a) Object detection for traffic lights, bicycles, yellow pavements **(b)** detection of road markings

Table 2: Traffic sign classifier performance

Traffic Sign	Precision	Recall	F-1 score
Pedestrian Crossing	0.99	1.00	0.99
Sidewalk closed	0.71	1.00	0.83
Signal ahead	0.99	0.97	0.98
Slow	1.00	1.00	1.00
Stop	0.99	0.99	0.99
Stop Ahead	1.00	0.97	0.98

6. Conclusions and Future Work

In this project we developed a novel, comprehensive vision system for the visually impaired for indoor and outdoor navigation, coupled with scene understanding. The system is simple, fashionable, unobtrusive, and not noticeable as an assistive device. Common challenges like detecting traffic signs, hanging obstacles, crosswalks, moving obstacles, elevation changes and geolocalization are addressed using advanced perception capabilities, implemented on a low-power device. A user-friendly voice interface allows users to easily control and interact with the system. Following several hours of testing in Monrovia, CA, we are confident that this project addresses the most common challenges faced by the visually impaired.

Our future work will explore options to run multiple semantic image segmentation models simultaneously at a higher inference rate. At the time of this work, the OAK-D sensor was a Kickstarter project due to which we were unable to obtain multiple sensors. We plan to evaluate system performance with multiple OAK-D sensors simultaneously. We will also experiment with traditional point cloud methods to detect elevation changes using the Generation 2 DepthAI module and incorporate robust object tracking across frames for more accurate traffic analysis More importantly, based on the insights gained from [31], [32], [33], we are confident that in our future work we can eliminate completely the need for a laptop, and replace it with a mobile device such as Google Pixel 2 and a low-power computing edge device such as Nvidia Jetson or TX2, making the application extremely mobile.

References

[1] Global Data on visual impairments 2010, WHO, Feb. 2021. Accessed: Feb. 21, 2021. [Online]. Available: https://www.who.int/blindness/GLOBALDATAFINALforweb.pdf.

[2] P. Ackland, S. Resnikoff, and R. Bourne. World blindness and visual impairment: despite many successes, the problem is growing. In *Community Eye Health*, vol. 30, no. 100, Art. no. 100, 2017.

[3] R. R. Bourne *et al.* Magnitude, temporal trends, and projections of the global prevalence of blindness and distance and near vision impairment: a systematic review and meta-analysis. In *Lancet Glob. Health*, vol. 5, no. 9, Art. no. 9, 2017.

[4] R. R. Bourne *et al.* Global Prevalence of Blindness and Distance and Near Vision Impairment in 2020: progress towards the Vision 2020 targets and what the future holds.. In *Invest. Ophthalmol. Vis. Sci.*, vol. 61, no. 7, Art. no. 7, 2020.

[5] E. Munemo and T. Tom. Problems of unemployment faced by visually impaired people. In *Greener J. Soc. Sci.*, vol. 3, no. 4, Art. no. 4, 2013.

[6] Daily Life Problems Faced by Blind People. In *Daily Life Problems Faced by Blind People*, Feb. 25, 2021. https://wecapable.com/problems-faced-by-blind-people/ (accessed Feb. 25, 2021).

[7] Challenges blind people face when living life. In *Challenges blind people face when living life*, Apr. 15, 2019. https://www.letsenvision.com/blog/challenges-blind-people-face-when-living-life (accessed Feb. 25, 2021).

[8] H. T. V. Vu, J. E. Keeffe, C. A. McCarty, and H. R. Taylor. Impact of unilateral and bilateral vision loss on quality of life. In *Br. J. Ophthalmol.*, vol. 89, no. 3, Art. no. 3, 2005.

[9] L. Zhi-Han, Y. Hui-Yin, and M. Makmor-Bakry. Medication handling Challenges among Visually Impaired Population. In *Arch. Pharm. Pract.*, vol. 8, no. 1, Art. no. 1, 2017.

[10] C. C. Collins. Tactile television-mechanical and electrical image projection. In *IEEE Trans. Man-Mach. Syst.*, vol. 11, no. 1, Art. no. 1, 1970.

[11] L. Kay. An ultrasonic sensing probe as a mobility aid for the blind. In *Ultrasonics*, vol. 2, no. 2, Art. no. 2, 1964.

[12] P. B. Meijer. An experimental system for auditory image representations. In *IEEE Trans. Biomed. Eng.*, vol. 39, no. 2, Art. no. 2, 1992.

[13] J. M. Loomis. Digital map and navigation system for the visually impaired, *Unpubl. Manuscr. Dep. Psychol. Univ. Calif. St. Barbara*, 1985.

[14] V. Kulyukin, C. Gharpure, J. Nicholson, and S. Pavithran. RFID in robot-assisted indoor navigation for the visually impaired. 2004, vol. 2, pp. 1979–1984.

[15] S. Shoval, J. Borenstein, and Y. Koren. The Navbelt-A computerized travel aid for the blind based on mobile robotics technology. In *IEEE Trans. Biomed. Eng.*, vol. 45, no. 11, Art. no. 11, 1998.

[16] I. Ulrich and J. Borenstein. The GuideCane-applying mobile robot technologies to assist the visually impaired. In *IEEE Trans. Syst. Man Cybern.-Part Syst. Hum.*, vol. 31, no. 2, Art. no. 2, 2001.

[17] M. Bousbia-Salah, A. Redjati, M. Fezari, and M. Bettayeb. An ultrasonic navigation system for blind people, 2007, pp. 1003–1006.

[18] K. Ito *et al.* CyARM: an alternative aid device for blind persons, 2005, pp. 1483–1488.

[19] D. Dakopoulos and N. G. Bourbakis. Wearable obstacle avoidance electronic travel aids for blind: a survey. In *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 40, no. 1, Art. no. 1, 2009.

[20] S. Real and A. Araujo. Navigation systems for the blind and visually impaired: Past work, challenges, and open problems. In *Sensors*, vol. 19, no. 15, Art. no. 15, 2019.

[21] R. Tapu, B. Mocanu, and T. Zaharia. A computer vision system that ensure the autonomous navigation of blind people, 2013, pp. 1–4.

[22] H. Jabnoun, F. Benzarti, and H. Amiri. Visual substitution system for blind people based on SIFT description, 2014, pp. 300–305.

[23] S. Meers and K. Ward. A substitute vision system for providing 3D perception and GPS navigation via electro-tactile stimulation, 2005.

[24] A. Rodríguez, J. J. Yebes, P. F. Alcantarilla, L. M. Bergasa, J. Almazán, and A. Cela. Assisting the visually impaired: obstacle detection and warning system by acoustic feedback, *Sensors*, vol. 12, no. 12, Art. no. 12, 2012.

[25] L. A. Johnson and C. M. Higgins. A navigation aid for the blind using tactile-visual sensory substitution, 2006, pp. 6289–6292.

[26] *DepthAI: Embedded Machine learning and Computer vision api*. Luxonis.

[27] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks, 2018, pp. 4510–4520.

[28] A. Rosebrock. Traffic Sign Classification with Keras and Deep Learning. In *Traffic Sign Classification with Keras and Deep Learning*, 11/4/12019. https://www.pyimagesearch.com/2019/11/04/traffic-sign-classification-with-keras-and-deep-learning/ (accessed Feb. 25, 2021).

[29] Intel. semantic-segmentation-adas-0001, Feb. 25, 2021. https://docs.openvinotoolkit.org/2019_R1/_semantic_segmentation_adas_0001_description_semantic_segmentation_adas_0001.html (accessed Feb. 25, 2021).

[30] Intel. road-segmentation-adas-0001, Feb. 25, 2021. https://docs.openvinotoolkit.org/2019_R1/_road_segmentation_adas_0001_description_road_segmentation_adas_0001.html (accessed Feb. 25, 2021).

[31] A. Rosebrock. In *Deep Learning for Computer Vision with Python Starter Bundle*, 3rd ed.

[32] Ignatov, A., Timofte, R., Kulik, A., Yang, S., Wang, K., Baum, F., ... & Van Gool, L. AI benchmark: All about deep learning on smartphones in 2019. In *2019 IEEE/CVF International Conference on Computer Vision Workshop*

pp. 3617-3635.

[32] Ignatov, A., Timofte, R., Chou, W., Wang, K., Wu, M., Hartley, T., & Van Gool, L. AI benchmark: Running deep neural networks on android smartphones. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*. pp. 0-0.

[33] AI Benchmark. Detailed ranking processors. https://ai-benchmark.com/ranking_processors_detailed.html (accessed Apr. 18, 2021).