IoT Assistant for People with Visual Impairment in Edge Computing

Manoel José de Souza Júnior,
Horácio Antonio Braga Fernandes de Oliveira and
Raimundo da Silva Barreto

January 25, 2021

IoT Assistant for People with Visual Impairment in Edge Computing

Manoel José S. Júnior[1], Horácio F. Oliveira[1], and Raimundo S. Barreto[1]

Institute of Computing - Federal University of Amazonas, Manaus, Brazil
{manoel.junior,horacio,rbarreto}@icomp.ufam.edu.br

Abstract. We advocate that technology can make it possible to develop devices capable of recognizing people and objects, mainly with the aid of machine learning, computer vision, and cloud computing. Such devices can be used in the daily life of a visually impaired person, providing valuable information for guiding their steps, providing a better quality of life. This paper proposes an architecture that uses computer vision and applies deep learning techniques to an Internet of Things (IoT) assistant for people with visual impairment. Considering that an IoT device is a limited device, it's used edge computing to improve the proposed architecture so that the device may be updated over time. The recognized object is converted into Text To Speech (TTS), allowing the user to listen to what has been recognized and also the distance from the user to the object. Unrecognized objects are sent to the cloud, and the device receives a re-trained network. The proposed architecture has been implemented using known and proved technologies such as Raspberry Pi 3, USB camera, Ultrasonic Sensor module, You Only Look Once (YOLO) algorithm, Google-TTS, and Python. Experimental results demonstrate that our architecture is feasible and promising.

Keywords: IoT · Machine Learning · Edge Computing · Computer Vision.

1 Introduction

Accessibility is the design of products, devices, services, or environments for people with disabilities in such a way that it guarantees the safety and physical integrity of people with special needs or reduced mobility, thus ensuring the right to come and go, and even to enjoy the same environments as a person without special need [3]. Accessibility usually can be viewed as the ability to access, but focused on enabling access for people with disabilities, or special needs, or enabling access through the use of assistive technology.

Recent advances have changed the way humans have lived. Computational devices have evolved to provide a better user experience in all areas of knowledge. However, there is a class of people who can benefit from technology to increase the quality of life and thus to live a healthy, normal life.

According to [1], it is predicted that there would be about 38.5 million people blind in 2020 (out of a total global population of 7.7 billion), equivalent to approximately 5 percent of the population. Most of these people live in confinement, without interaction with the outside world, because they can not move around due to the absence of adequate support.

Despite the existence of several technologies, such as GPS and ultrasonic sensor, that enable a walking stick to measure the distance to objects, there is an opportunity to improve accuracy detection and inform what is the object type ahead of the user, such as a person, a dog, a hole, and so on.

The Internet of Things, or IoT, is a system of interrelated computing devices, objects, animals or people that are provided with unique identifiers and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction. This includes everything from cellphones, glasses, coffee makers, washing machines, headphones, lamps, wearable devices, and almost anything else you can think. Nowadays, IoT devices are capable of performing sophisticated detection and recognition tasks to provide interactions between humans and their physical environment.

The inclusion of Deep-learning on IoT devices often demands real-time requirements. For example, a security camera that does object-recognition tasks usually requires a detection latency of less than 500 ms to capture and respond to target [14].

Considering that an IoT device is a limited device, it's used edge computing to improve the proposed architecture, in such a way that the device can be updated over time. The recognized object is converted into speech, allowing the user to listen to what has been recognized and also the distance from the user to the object. Unrecognized objects are sent to the cloud, and the device receives a re-trained network.

The remainder of this paper is organized as follows. In the next section, the related work is presented. Our proposed system is then explained in section 3. We show and discuss the implementation of the system in Section 4. The results and future work are described in section 5. Finally, in section 6 the conclusions and future work is presented.

2 Related Work

There are other projects aimed to help people with visual impairment. For instance, the approach described by Choudhury et al. in [6] shows the transformation of images in Braille using You Only Look Once (YOLO) algorithm. However, there are other faster ways to guide a person without the need to interpret signals. In [12], a prototype uses algorithms to detect real-life symbols, such as a bathroom, a subway, or a snack bar. However, it does not help the user reaching these locations safely during the route. Several objects and obstacles can cause accidents in your percussion. In contrast, Alam et al. used various technologies to find the location within a closed environment, but visually impaired people need devices that enable their interaction with the outside environment [7].

In our proposed architecture, the glasses allow an outdoor experience, giving more freedom and quality of life to the user of the device. It is also possible to receive guidance quickly without the need for interpretation. This work also adds value to the work [12], being able to find the obstacles present in the path of the visually impaired.

Therefore, in comparison with the above studies, it is possible to observe that much work has already been done. However, what highlights this project is that this proposal aims to identify the distance of objects, then converting video in real time in speech format (TTS). Also, objects present in the user route that was not identified is sent to the cloud. In the cloud, a neural network retrains the model and returns an update of the software, so the device becomes more robust and customized to the user. All of this is possible using resources for IoT, Machine Learning, and Cloud Computing. In the next section, the proposed model is discussed in details.

3 System Proposed

The model presented allows the use of two sensors: an ultrasonic sensor to measure the distance and the camera to capture images.

The captured distance is converted into voice, using Google Text to Speech (TTS). The photo captured by the camera goes through a resizing process to be inserted in an already trained network that receives the photo and goes through a classification process using the YOLO algorithm. YOLO is a neural network-based object detection algorithm. It receives an image as input and returns another picture with boxes around possible objects known to the already trained network. These objects then receive a label. Joseph Redmon created an implementation of this study, called darknet, developed in C language and CUDA [2].

The model uses this already trained network that covers a variety of objects. To better present the strategy, one unidentified object was used. Because it is impracticable to train this network in computational constrained IoT devices, the concept of edge computing is used, allowing near, local, cloud-based devices to take care of the processing job.

He Li has demonstrated in his work that it is possible to implement machine learning on IoT devices with the help of edge computing to share computing resources, not overloading devices since they have processing, memory and power limitations [8]. Based on this principle, the unidentified object can be shared with a PC with greater processing power (edge device) and, then, this object goes through a process of deep learning matching.

The most widely used neural networks for processing deep learning workloads are Convolutional Neural Network (CNNs), which convert unstructured image data into structured object label data. Generally, CNNs work as follows: first, a convolution layer scans the input image to generate a feature vector; second, an activation layer determines which feature in the vector must be activated for the image under inference; third, a layer of pooling reduces the size of the resource vector [14].

The training result is a file with changed weights. After finishing the training, the device is notified that there is an update to be made. When the user accepts the update, the device can identify the previously unidentified object.

4 Implementation of the system design

4.1 Architecture

In Fig. 1 it is possible to see the division of architecture in 3 main parts. The first part is the input of the system. A Raspberry-based device represents the second part of system (the software components used in the design), and the third part is the output of the system.

The system receives as input the images provided by the USB camera, which is represented by the label 1 in the figure. The distance of the nearest object is then informed by the ultrasonic sensor represented by label 2 of the figure.

In the center of the image, represented by the label 3, is the Raspberry device. The implementation uses the *Raspbian version 2019-04-08-raspbian-stretch-full*, this version contemplates Python v3 by default. Python was used to create the state machine and to do the integration of the modules. *Numpy*, *Pandas*, and *Keras* is used to do some manipulation and testing. Also, the *Jupyter notebook* had a key role in the information sharing to simulate edge computing.

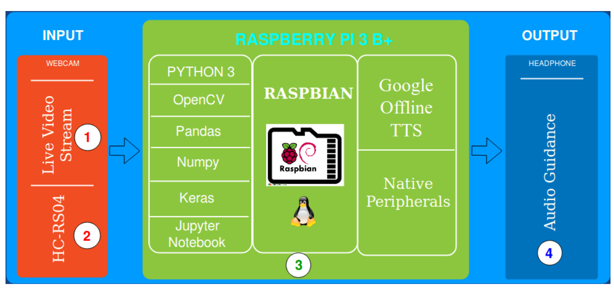

Fig. 1. Block Diagram.

4.2 Software Design

The state machine developed to provide the user with the option of choosing a mode of use, it is represented by a finite automaton, where each state represents an option, as shown in Fig. 2.

(a) *Instruction*: represented by the symbol *q0*, the device initiates in it when it is turned on, we call initial state. Here the device speaks the available options, giving the user the options to choose which mode to use.

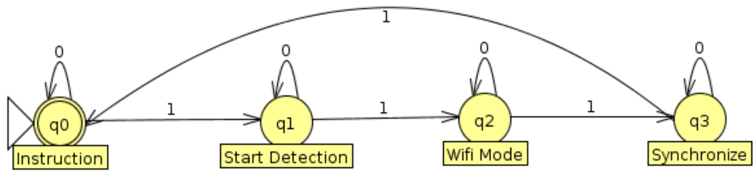

Fig. 2. State machine of system

(b) *Start Detection*: this state, represented by the symbol *q1*, is responsible for triggering the camera to capture the frames for analysis of the algorithm and to measure the distance of the objects. In the next, step the detection algorithm is started. If the object is recognized, the object name and distance from the nearest object is sent to the speech synthesizer. If the object has not been identified, the image is stored, and the speech synthesizer only receives the distance from the object. Even though the object has not been identified, the user is informed that there is an unidentified object and how far away it is, as shown in Fig. 3.

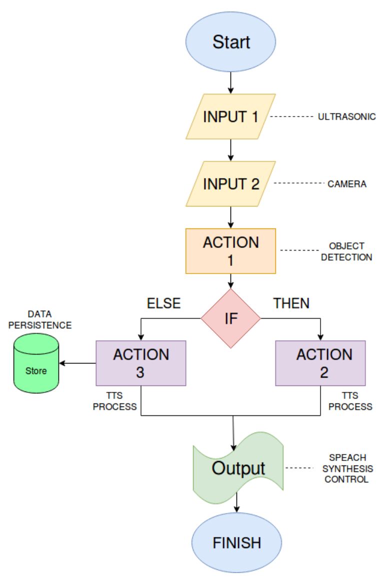

Fig. 3. Detection Cycle

(c) *Wi-Fi Mode*: this state, represented by the symbol *q2*, is the mode of connectivity, responsible for searching for wireless networks. The device identifies which Wi-Fi networks are available and uses the text to Speach, to indicate the SSIDs available for the user to connect manually. SSID is the acronym for "service set identifier." The choice for a WI-FI network and the password, were defined directly from the code at the time of implementation. To make this feature available to the user, we map a strategy that is described in the section 5

(d) *Synchronize*: represented by the symbol *q3*, if the device is connected to the internet, in this state, the unidentified images are sent to the server. The cloud or receive neural network updates, ensuring synchronization (see Fig. 4).

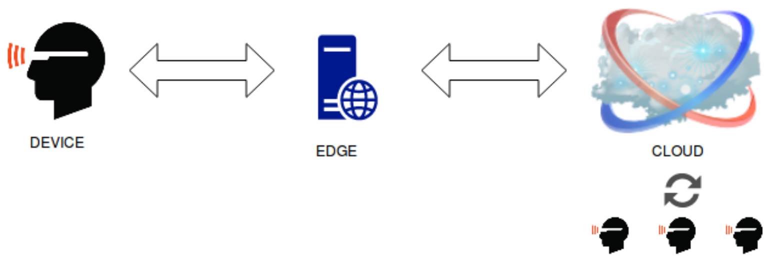

Fig. 4. Synchronize

If the device never connects to the network, it will not be possible to add new objects to be recognized. In this way the architecture would not work and not reach the expected goal.

Another key point that is worth highlighting: the proposed architecture allows communication and cloud storage, which is to say if we have multiple such devices around the world, they can share the experience of the other user. This also allows the device that needs to upgrade, does not necessarily need to retrain its network, because in the cloud, there may already be a request previously made by another user.

4.3 Hardware Design

Raspberry Pi 3 Model B + is a mini-PC that runs Linux distributions like Raspbian and Ubuntu but also supports other operating systems like Windows 10 IoT and custom versions of Linux. It has a BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz Broadcast processor. 1GB of LPDDR2 SDRAM. 2.4GHz and 5GHz IEEE 802.11b/g/n/ac Wireless LAN, Bluetooth 4.2, BLE.

Was used a Logitech C270 HD model camera for live streaming. A powerful feature of the Raspberry Pi is the GPIO (general purpose input/output) pin line along the top edge of the board. A 40-pin GPIO header is found on all current Raspberry Pi cards [4]. The GPIO was used to add buttons and connect the ultrasonic sensor. Although raspberry provided input and output (IO) interfaces, it was necessary to develop hardware to feed the ultrasonic modules separately and add resistors because of the voltage difference. It is possible to see in Fig. 5 the developed circuit board.

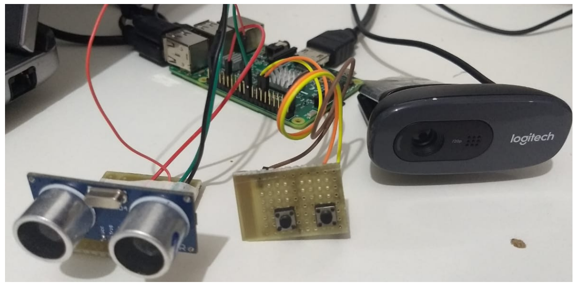

Fig. 5. First prototype

In Fig. 6, there are two buttons. The first button allows you to select the usage mode (*Instruction*, *Start Detection*, *Wi-fi Mode*, or *Synchronize*). The second button is the confirmation button. To improve usability, braille translations are placed next to the buttons.

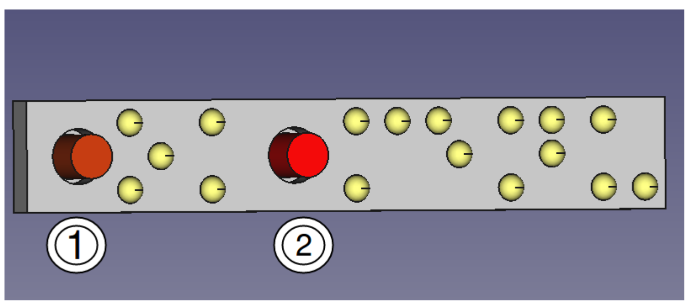

Fig. 6. Buttons with Braille translations.

5 Results and Future Work

5.1 Accuracy test

In this section, the architecture validation is showed. The first experiment was to perform the detection algorithm on an object known by the network. The time for this operation was 29 seconds, with 93% accuracy.

Then the same test was performed, but for an unidentified object. The time for this operation was 33 seconds. We noticed that when the object is not identified, the network takes a little longer to reply.

With the unidentified object collected, was calculated the time to send the image to the edge with a resolution of 600x480 pixels. The time spent on sending the image was of $4s$. The quality of the connection at the time of the upload was $7.02Mbps$.

With the image at the edge, the training time was 5 minutes, taking into account the configuration of the device with GPU of 12GB-RAM.

Returning re-training network, the time to upgrade the device was 4 minutes, for a 300MB size file, and the quality connection at that time was $9.95Mbps$ for download. The results can be seen in Fig. 7.

TEST	Time	Accuracy
Known object	29 seconds	93%
Unknown object	35 seconds	0%
Send image to Edge	4 seconds	
Download network weights	4 minutes	
New object added	30 seconds	80%

Fig. 7. Test Result.

On the left of Fig. 8, the photo of the eagle is an example of the system, at the opposite end we have a hole in the middle of the street, this was the image used to retrain the network.

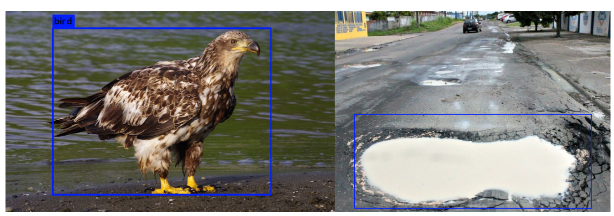

Fig. 8. Comparison.

We also did distance experiments based on the specification of the ultrasonic sensor. Its range of action is of $2cm$ to $4m$. It was observed with some experiments ranging from 1 to $4m$, As can be seen in Fig. 9

The accuracy testing of the YOLO algorithm shows 80 to 100% accuracy for objects recognized by the network. With the addition of new objects, accuracy remained at the same margin of success.

5.2 Problems occurred

Since we did not have a computer with good performance to retrain the network in edge, we had to use the cloud to simulate this environment. Another problem

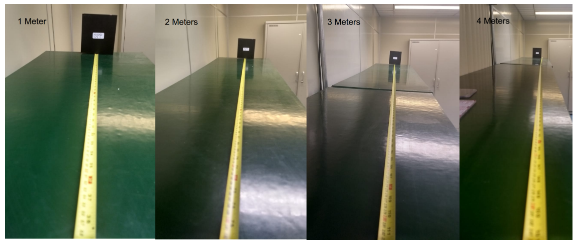

Fig. 9. Ultrasonic module accuracy test.

was encountered during developed the prototype. The response time was between 12 to 14*min* to identify an object. It was circumvented the problem by using the raspberry GPU, decreasing response time considerably to 10 to 30*s*. With this result, it was identified that the optimization of the operating system, can improve even more the response time.

5.3 Future Work

As a future work, we will implement voice recognition to integrate with *TTS* allowing the user can not only know the available *Wi-Fi* networks, but he can also speak the name of the network to which he wants to connect, desired network password.

6 Conclusion

This paper examined an architecture that uses computer vision and applies deep learning techniques to an IoT device.

The combination of technologies has shown that some substantial resources can be shared to achieve a common good.

He also showed that the advancement of technology could benefit people with visual limitations, bringing a better quality of life and interaction with the outside world.

Also, it has been demonstrated that these smart devices can be improved over time and adaptable to the routine of their users, making the device unique and personal.

All the tests performed presented good experimental results, thus demonstrating that our architecture is feasible and promising.

7 Acknowledgement

This research was partially supported by Priority Program for the Training of Human Resources - CAPDA / SUFRAMA / MDIC, under the terms of Federal Law n 8.387/1991.

References

1. The Lancet Global Health, https://www.thelancet.com/journals/langlo/article /PIIS2214-109X(17)30293-0/fulltext. Last accessed 19 Apr 2019.
2. Joseph Chet Redmon https://pjreddie.com/darknet/yolo/. Last accessed 18 Apr 2019.
3. United Nations - Convention on the Rights of Persons with Disabilities https://www.un.org/development/desa/disabilities/convention-on-the-rights-of-persons-with-disabilities.html. Last accessed 21 May 2019.
4. Raspberry Organization https://www.raspberrypi.org/documentation/usage/gpio/ Last accessed 23 Apr 2019.
5. T. A. Heya, S. E. Arefin, A. Chakrabarty and M. Alam, ”Image Processing Based Indoor Localization System for Assisting Visually Impaired People,” 2018 Ubiquitous Positioning, Indoor Navigation and Location-Based Services (UPINLBS), Wuhan, 2018, pp. 1-7.
6. A. A. Choudhury, R. Saha, S. Z. Hasan Shoumo, S. Rafsun Tulon, J. Uddin and M. K. Rahman, ”An Efficient Way to Represent Braille using YOLO Algorithm,” 2018 Joint 7th International Conference on Informatics, Electronics & Vision (ICIEV) and 2018 2nd International Conference on Imaging, Vision & Pattern Recognition (icIVPR), Kitakyushu, Japan, 2018, pp. 379-383.
7. M. M. Alam, S. E. Arefin, M. A. Alim, S. I. Adib and M. A. Rahman, ”Indoor localization system for assisting visually impaired people,” 2017 International Conference on Electrical, Computer and Communication Engineering (ECCE), Cox's Bazar, 2017, pp. 333-338.
8. H. Li, K. Ota and M. Dong, ”Learning IoT in Edge: Deep Learning for the Internet of Things with Edge Computing,” in IEEE Network, vol. 32, no. 1, pp. 96-101, Jan.-Feb. 2018. doi: 10.1109/MNET.2018.1700202
9. Feng Lan, Guangtao Zhai and Wei Lin, ”Lightweight smart glass system with audio aid for visually impaired people,” TENCON 2015 - 2015 IEEE Region 10 Conference, Macao, 2015, pp. 1-4.
10. Z. M. Fadlullah et al., ”State-of-the-Art Deep Learning: Evolving Machine Intelligence Toward Tomorrows Intelligent Network Traffic Control Systems,” in IEEE Communications Surveys & Tutorials, vol. 19, no. 4, pp. 2432-2455, Fourthquarter 2017.
11. M. Verhelst and B. Moons, ”Embedded Deep Neural Network Processing: Algorithmic and Processor Techniques Bring Deep Learning to IoT and Edge Devices,” in IEEE Solid-State Circuits Magazine, vol. 9, no. 4, pp. 55-65, Fall 2017.
12. K. R. Rani, ”An audio aided smart vision system for visually impaired,” 2017 International Conference on Nextgen Electronic Technologies: Silicon to Software (ICNETS2), Chennai, 2017, pp. 22-25.
13. S. Yao et al., ”Deep Learning for the Internet of Things,” in Computer, vol. 51, no. 5, pp. 32-41, May 2018.
14. J. Tang, D. Sun, S. Liu and J. Gaudiot, ”Enabling Deep Learning on IoT Devices,” in Computer, vol. 50, no. 10, pp. 92-96, 2017.
15. Mohammad Saeid Mahdavinejad, Mohammadreza Rezvan, Mohammadamin Barekatain, Peyman Adibi, Payam Barnaghi, Amit P. Sheth,Machine learning for internet of things data analysis: a survey,Digital Communications and Networks,Vol 4, no. 3,2018,pp. 161-175.
16. P. Mendki, ”Docker container based analytics at IoT edge Video analytics usecase,” 2018 3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT-SIU), Bhimtal, 2018, pp. 1-4.