Real-time object detection aid for the visually-impaired

Alisa Levin and Rahul Mitra
Advisor: Peter Yoon
Department of Computer Science
Trinity College, Hartford, CT, USA
email: {alisa.levin, rahul.mitra}@trincoll.edu

***Abstract*— 2.2 billion individuals world-wide suffer from visual impairment. Current technology-based aids are expensive and inaccessible. In our project, we leveraged advances in microcomputer technology to create an efficient, easy-to-use, real-time object detection system for the visually impaired. This system consists of a Jetson Nano microcomputer which receives video input from an attached camera. The Nano uses a deep learning model to identify objects in the user's immediate environment in real-time. This information is then relayed to the user auditorily through an iOS smartphone application that interfaces with the Nano. To achieve this, we implemented a pre-trained object detection model, SSD Inception v2, as well as a Bluetooth GATT server on the Nano. We developed an iOS application in order to facilitate the real-time wireless transfer of this model's classification labels from the Nano to the user's smartphone. The system was mounted on a white cane to enhance a tool already commonly used. We hope our system is a step towards an open source and affordable alternative to current visual aid technologies.**

***Keywords* — Jetson Nano, real-time classification, Bluetooth, SSD Inception v2, iOS application**

I. Introduction

According to the World Health Organization, at least 2.2 billion people are visually impaired or blind worldwide [1]. Real-time object detection and identification is an everyday challenge faced by such individuals. Trouble successfully detecting and identifying objects hinders an individual's efficiency and quality of life. Consider a hypothetical situation where a visually impaired individual is at a busy crossing. This individual will likely experience a combination of auditory, tactile, smell and potentially partial visual feedback. Since the visually impaired rely heavily on sounds and touch to identify objects, this particular scenario might result in a sensory overload for the individual that makes identification of the objects around them very challenging. To address this, we present a convenient and fast system that identifies, in real-time, objects in a visually impaired person's field of vision. Once object identification is achieved, our system relays information about the objects to the individual via audio.

While many real-time object detection methods exist [2]–[4], even the most sophisticated algorithms tend to suffer from high computational cost [5]. Many advanced object detection techniques [6]–[8] rely heavily on specialized hardware architectures and powerful computers. Implementing these methods on ubiquitous, portable devices such as smartphones is computationally unrealizable due to hardware constraints. As such, while efficient, these techniques are not portable or practical for everyday usage. To overcome these portability and practicality issues, our system uses an NVIDIA Jetson Nano microcomputer and a Raspberry Pi camera for object detection and identification. The microcomputer, after having done all the processing necessary for object identification from the camera's input, sends information to an intuitive smartphone application. The smartphone application, using swipe-controlled audio feedback, relays information about the individuals surroundings. Finally, a white cane is a navigation tool that numerous visually impaired individuals have on their body at all points in time. To avoid extraneous hardware, we configured our system with a white cane as shown in Fig. 1. Any time an impaired individual has the cane on their body, they will, through our smartphone application, be able to identify objects in their immediate surroundings. This could aid them in their navigation of environments crowded with traffic, passersby, and more.

Fig. 1: System configured with white cane.

II. Implementation

Our system required two distinct components. First, a machine learning real-time object detection system was implemented on the Jetson Nano microcomputer. Second, an application to interface with the Nano was implemented on an iOS smartphone.

A. Jetson Nano

The Jetson Nano is equipped with the following hardware components:

i Raspberry Pi Module v2 camera that captures video input.

ii INIU Portable Battery that supplies approximately 2 hours of continuous power.

iii Intel 8265NGW Bluetooth/Wifi Module that incorporates Bluetooth functionality into the Nano. We chose to implement a Bluetooth connection so that the user may interact with the application, irrespective of whether a Wifi connection is available or not.

The entire system is mounted on white can as shown in Fig. 2.

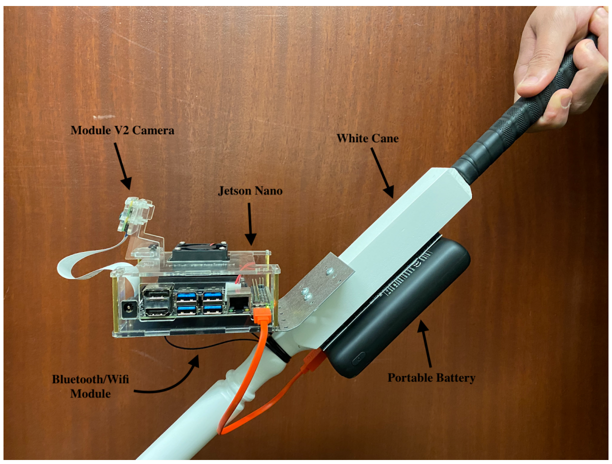


Fig. 2: Jetson Nano equipped with hardware components.

After setting up the hardware, the Nano needed to detect objects in real-time, maintain a logical Bluetooth connection initiated by the smartphone and use that connection to send information about the identified objects to the iOS application on the smartphone. To achieve these goals, we used the following software components:

i **SSD Inception v2** which was our model of choice for real-time object detection. It has 91 object classes.

ii **BlueZ**, a Linux Bluetooth protocol stack, was used to set up a logical Bluetooth GATT server on the Nano.

iii **DBus**, a Linux interprocess communication system, was used to allow the machine learning process to continuously communicate with the Bluetooth GATT server on the Nano.

Choosing an appropriate machine learning model proved to be a challenging process. We experimented with YOLOv3 [9], which is a much larger network when compared to SSD Inception v2. However, due to hardware constraints, we could not consistently use YOLOv3 for object-detection on the Nano. As such, we continued to use SSD Inception v2 for our work.

Another challenge we faced throughout the development process was the unexplainable overheating of our Jetson Nano. Sometimes, the microcomputer would function perfectly well. At other times, it would heat up and crash after only a few minutes of use. This was very frustrating, as it greatly slowed down the testing of our system as we built upon our code over time. Though adding a fan to the Nano's heat sink helped mitigate the issue slightly, our search for common factors that could reveal the root of the Nano's overheating problem was unsuccessful. However, this overheating did not significantly affect the performance of our final system.

Bluetooth low energy (BLE) protocol uses the central-peripheral model of communication. In most documentation, the Nano is used as a central device i.e., it initiates and controls a Bluetooth connection with a peripheral device, often receiving data from it. However, in our project, we needed to configure the Nano as a peripheral device i.e., have it wait for a smartphone to initiate a Bluetooth connection before sending over data to the central device. To achieve this, we spent multiple weeks researching various methods of setting up a Bluetooth peripheral on a Linux operating system. Eventually, we came across a solution that was successful: we used BlueZ to implement a python Bluetooth GATT server on the Nano [10]. This allowed us to send data from the Nano to the iOS application.

B. Smartphone Application

The smartphone application, Object Detection Aid (ODA) uses the following software features:

i **Core Bluetooth**, a popular Swift BLE library was used to interface with the Nano. A logical Bluetooth connection is set up on the iOS application. This Bluetooth connection receives detected objects, encoded as integer IDs, from the Nano.

ii **Look-up table** that maps integer IDs to string labels allows our application to convert from ID to detected object label.

iii **Audio feedback** is provided using the above label. The AVSpeechSynthesizer library was used to auditorily relay the identified objects to the user.

Our system workflow is described in Fig. 3.

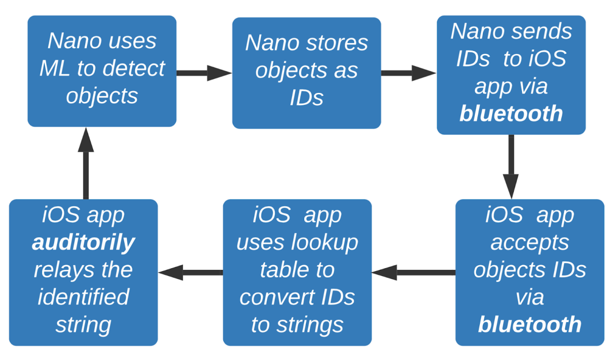


Fig. 3: Interaction between Nano and iOS application.

Our primary focus in developing this application was functionality, which we achieved considerably well. The major challenges we had to overcome were common issues faced when picking up a new programming language such as syntax, variable bindings, scope, available data structures etc. However, the popularity of Swift ensured we had a lot of available documentation and examples to refer to, which greatly helped in the construction of our application.

C. White Cane

To integrate this system with a white cane, we assembled a plastic case around Jetson Nano, which included a cooling fan for the heat sink. This served to protect the microcomputer as we mounted it onto a metal platform that was screwed to the white cane at a 130 degree angle. Finally, we secured the battery to the back of the white cane, directly behind the Nano. For this, we used heavy-duty Velcro, to ensure that the battery could be detached for convenient recharging.

III. Results & Outcomes

Our project was successful in achieving its initial goals. On the Jetson Nano, we were able to implement the **SSD Inception v2** model to identify objects in real-time as shown in Fig. 4. Objects were stored as encoded integer IDs.

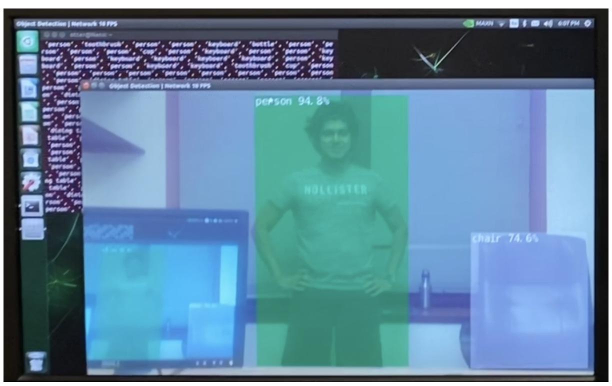
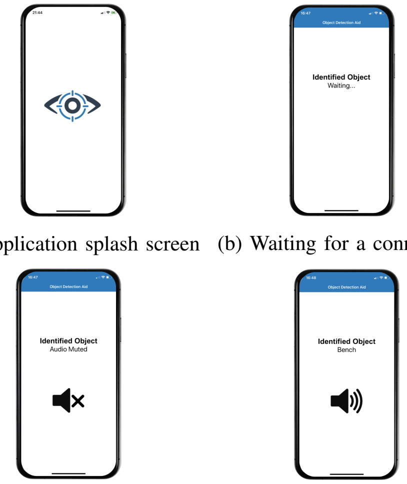


Fig. 4: Bounding boxes signify identified objects

On the smartphone, we were able to use **Core Bluetooth** to successfully implement a Bluetooth connection with the Jetson Nano. The application received IDs encoded as integers via the Bluetooth connection. The integers were then mapped to string labels. We used the *AVSpeechSynthesizer* iOS package to auditorily relay these labels to the user in real-time. The behaviour of our application is detailed in Fig. 5. Fig. 5(a) shows the splashscreen when the user first opens the application. This screen displays the application logo. In Fig. 5(b), the application is waiting for a Jetson Nano to connect via Bluetooth. When the user is at the stage described by Fig. 5(b), the application also provides audio feedback by relaying the sentence, ”Waiting for connection”. In Fig. 5(c), the application has registered a connection with the Nano but the user has performed a left-swipe. As such, the audio muted symbol and message is displayed. Finally, in Fig. 5(d), the user has performed a right-swipe



(a) Application splash screen (b) Waiting for a connection

(c) Left swipe - audio muted (d) Right swipe - audio on

Fig. 5: Behaviour of the *Object Detection Aid (ODA)* application

on the screen. The application now provides audio feedback about the user’s immediate environment. In particular, the application in Fig. 5(d) detects a bench and the string, ”bench” is auditorily relayed.

A. Project Features

Many accessibility devices for disabled individuals have been digitalized in recent years to improve their efficiency and effectiveness. Despite this, the white cane, which was invented nearly 100 years ago, remains a simple, manual navigation aid for the blind [11]. Just this past year, a startup called WeWALK released a smart cane for the blind that can detect obstacles and serve as a GPS navigation tool through audio and haptic feedback. However, this is a very expensive tool, marketed at $599, and is therefore inaccessible to the majority of visually impaired individuals. In fact, visual impairment across the world is four times as prevalent in middle and low-income regions [1], making it all the less practical for such a beneficial tool to be so costly. To combat this, our project leverages affordable, easy-to-use technologies to create an accessible aid for visually impaired individuals. In fact, the total cost of the materials we used is $245, just 41% of the cost of the WeWALK cane. It is important to note that this amount arose from us buying each individual component at retail cost, making the expense of our system much greater than if we were to develop it using mass-produced, wholesale components. Thus, our aid has the potential to become an even more affordable alternative solution.

Reviewers of the WeWALK cane also find its constant additional haptic feedback more frustrating than helpful, as it only accentuates the natural haptic feedback of the cane moving across the ground [12]. To avoid this pitfall, our design does not add any haptic feedback; users can instead easily turn the audio feedback from the smartphone on or off at their convenience, while the natural, unaltered haptic feedback of the white cane remains. Furthermore, the WeWALK cane is limited by the fact that it can only alert users of the presence of ambiguous obstacles, but not actually identify them. In contrast, by incorporating a camera onto the white cane and processing its video via machine learning, our aid is able to identify many different obstacles, alerting the user of which exact objects are in their field of view.

An additional advantage of our design is that it is open-source, with a flexible codebase that can be adapted for various purposes. For example, machine learning models trained on different datasets could be added to the code for more specialized use-cases, from traveling on public transport in the city to clothes shopping at the mall. Similarly, within practical hardware constraints, this system can be implemented on any microcomputer, with any camera, using any suitable battery, and be mounted on any sturdy white cane. This flexibility of implementation clearly demonstrates the accessibility and affordability of our solution.

B. Limitations

Although our design is an efficient and effective proof of concept, it still has some limitations. The object detection model we used, SSD Inception v2, only detects 91 classes of objects, many of which are not useful for our application; although "stop sign", "fire hydrant" and "car" are relevant objects to identify while walking down the street, "broccoli", "toothbrush" and "teddy bear" are not. Since this model is not specialized, there is also a disparity between its performance on classifying some objects versus others, almost as if it was not trained on a stratified dataset. For example, the model can detect cars extremely well as they drive down the street, and is similarly successful at identifying people in motion. However, it struggles much more with identifying other objects even at a standstill, such as street signs and benches.

Another limitation of our design is that the camera mount on the Nano's case is non-adjustable. This results in an awkward orientation of the camera that cannot adapt to the various angles at which users may hold the white cane as they walk, and consequently leads to many missed object detections. Additionally, with all of the hardware attached to the white cane—including a particularly bulky battery—the cane also weighs quite a lot. This makes our cane, much like the WeWALK cane, too heavy for comfortable all-day use.

For our proof of concept, we chose to develop an iOS application. However, iOS only owns 27% of the world's mobile operating system market share, while Android owns 72% [13]. Thus, to improve the accessibility of our aid, we would need to create an Android version of our ODA application. Furthermore, it is important to note that not all people can afford smartphones in the first place. Therefore, an ideal solution would involve creating an application that is compatible with any type of central device with Bluetooth capabilities.

Finally, a significant limitation of this project is our lack of feedback and insights from visually impaired individuals. It would have been very helpful to speak to people within this target population directly at the beginning of our project design process in order to ensure that we were developing a useful aid that actually met a need within the visually impaired population or mitigated a shortcoming of currently available assistive technologies. It would have been valuable to then maintain a dialogue with these individuals, receiving their iterative feedback until the completion of our project. Unfortunately, we realized this shortcoming too late, and were unsuccessful in reaching out to online accessibility communities for advice and feedback. This experience taught us the importance of gathering feedback from end-users from the very start of the design process.

IV. Conclusion & Future Considerations

Visual impairment affects a large fraction of individuals. Technology-based aids, while available, are costly and inaccessible, defeating their purpose as a large-scale solution. In our project, we use an NVIDIA Jetson Nano microcomputer to produce a cost-effective and efficient aid that will hopefully assist the visually impaired. We present a real-time object detection system which relays auditory feedback about the user's immediate environment using an iOS application that interfaces, via Bluetooth, with the Nano. Our entire system is mounted on a white cane, to enhance a tool already commonly used by the visually impaired community.

The primary advantages of our system include portability of software, cost-effectiveness and object identification as opposed to current designs, which only relay the presence of objects. Our codebase is open-source and amenable to any NVIDIA architecture. As mentioned in Sec. III, the total cost of our design was only $245, and could be further reduced by using wholesale parts. Finally, unlike the WeWALK smartcane, our system actually informs the user of the exact object in their immediate environment. We are optimistic that these features allow our system to meet the needs of end-users. To view our code and a demo video, see: *https://github.com/rahul-mitra13/Object_Detection_Aid*.

Immediate future steps include replacing the rigid-angle camera mount with an angle-adjustable mount. This will help to improve detection rate. Another simple improvement would be to have a smaller, more lightweight battery power the Nano so that the cane is more suitable for extended periods of use. Further, we are hoping to use a larger object detection model that is more specialized to our use-case. As discussed in Sec. III, the SSD Inception v2, while effective for our prototype, is not built for our needs. In this prototype, the focus of the smartphone application was on functionality. Going forward, we hope to continue developing the application. In particular, we hope to improve the UI and incorporate additional features (such as a landing page, a user-guide, and more audio feedback customization options, etc.). Finally, the most significant future direction involves having our system tested by the individuals in the accessibility technology community, both end-users as well as domain experts. This community feedback will be invaluable in understanding how our work may be improved as well as additional features we may need to include.

V. Acknowledgements

We would like to thank all of the Trinity College Computer Science Department faculty for their guidance and support. We would also like to extend our gratitude to The Travelers Companies, Inc. for generously funding our project.

References

[1] J. D. Steinmetz, R. R. Bourne, P. S. Briant, S. R. Flaxman, H. R. Taylor, J. B. Jonas, A. A. Abdoli, W. A. Abrha, A. Abualhasan, E. G. Abu-Gharbieh *et al.*, "Causes of blindness and vision impairment in 2020 and trends over 30 years, and prevalence of avoidable blindness in relation to vision 2020: the right to sight: an analysis for the global burden of disease study," *The Lancet Global Health*, vol. 9, no. 2, pp. e144–e160, 2021.

[2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *arXiv preprint arXiv:1506.01497*, 2015.

[3] P. Viola, M. Jones *et al.*, "Robust real-time object detection," *International journal of computer vision*, vol. 4, no. 34-47, p. 4, 2001.

[4] B. Yang, W. Luo, and R. Urtasun, "Pixor: Real-time 3d object detection from point clouds," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 7652–7660.

[5] K.-H. Kim, S. Hong, B. Roh, Y. Cheon, and M. Park, "Pvanet: Deep but lightweight neural networks for real-time object detection," *arXiv preprint arXiv:1608.08021*, 2016.

[6] C. Huang and R. Nevatia, "High performance object detection by collaborative learning of joint ranking of granules features," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 41–48.

[7] C. Kyrkou and T. Theocharides, "A flexible parallel hardware architecture for adaboost-based real-time object detection," *IEEE Transactions on very large scale integration (VLSI) systems*, vol. 19, no. 6, pp. 1034–1047, 2010.

[8] ——, "A parallel hardware architecture for real-time object detection with support vector machines," *IEEE Transactions on Computers*, vol. 61, no. 6, pp. 831–842, 2011.

[9] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[10] F. J. Dian, A. Yousefi, and S. Lim, "A practical study on bluetooth low energy (ble) throughput," in *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*. IEEE, 2018, pp. 768–771.

[11] P. Strong, "The history of the white cane," 2009.

[12] M. Mulligan. (2020) Wewalk smart cane for the blind review - first impressions - (o&m specialist). Blind on the Move. [Online]. Available: https://www.youtube.com/watch?v=wwAcQx5bVmM

[13] (2020) Mobile operating system market share worldwide. Statcounter Global Stats. [Online]. Available: https://gs.statcounter.com/os-market-share/mobile/worldwide/2020