SightAid: Virtual Assistant for Vision Impaired

Abhishek Sonawane, Adarsh Hiremath, Arshdeep Sachdeva, Bala Kondaveeti, Dhrumil Joshi, Varad Deshmukh
{assonawl, ahirema3, asachd14, bkondave, djoshi12, vdeshmu4}@asu.edu
CSE 535: Mobile Computing, Arizona State University, Tempe, AZ

Abstract

In a harmonious blend of innovation and empathy, our application caters to the distinctive needs of the visually impaired. Rooted in a dual-feature architecture, the Braille keyboard seamlessly integrates physical gestures and buttons, offering effortless text generation. Simultaneously, the Object Detection module, driven by a robust Convolutional Neural Network, empowers users with real-time awareness of their surroundings through audio feedback. This transformative solution not only enhances daily experiences but also signifies a commitment to social good. With boundless potential in assistive technology, our project pioneers a new era of inclusivity, promising continuous refinement and empowerment for the visually impaired community. Find the complete implementation at https://tinyurl.com/SightAid-G20.

1. INTRODUCTION

Globally, at least 2.2 billion[1] people have vision impairment. In a world where sight is often taken for granted, these individuals face unique daily challenges. Our application is specifically designed to alleviate these daily struggles and enhance their quality of life. The motivation behind this project is the realization that, despite living in a tech-driven society, there needs to be more tailored tools available for these individuals that we hope to provide to them using our application. Here are the primary motivations behind this project:

- Challenges in Identifying Objects: The challenges in recognizing objects can significantly impact the mobility and safety of visually impaired individuals. The inability of smartphones to assist in these situations reduces their independence.
- Difficulty in Participating in Written Activities: Engaging with written content remains a significant challenge. In this study[2], where researchers tried teaching introductory statistics to blind students, they faced challenges such as expensive braille books and the inability to write the answers without proper tools.

Our project aims to bridge these gaps by leveraging smartphone sensors. The proposed application will be a Guardian Angel for the visually impaired individuals providing them an effective tool to understand their physical environment and also offering a comprehensive platform to express their ideas.

2. ARCHITECTURE

The application's architecture is centered around two key feature components: the Braille keyboard and the Object Detection module, designed to cater to the unique needs of visually impaired users.

The Braille keyboard incorporates a combination of physical buttons and gestures, processing user inputs to generate Braille intermediate, which is then queried against a Braille-to-English database for translation. This component allows users to effortlessly generate text in natural language, providing a versatile and accessible means of communication.

On the other hand, the Object Detection module utilizes the device's camera to capture images, sending them in a series of frames to a Convolutional Neural Network (CNN) module. This module, leveraging advanced computer vision techniques, generates labels for detected objects, converting the information into audio feedback for real-time awareness.

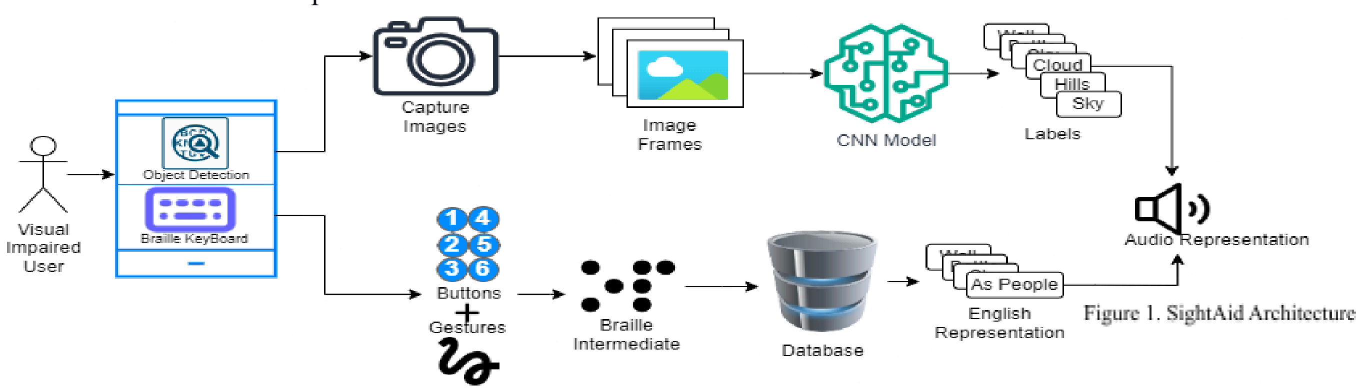

Figure 1. SightAid Architecture

3. THE APPLICATION SUITE

The application suite consists of the two components features of the project- Objection detection and Braille Keyboard. The objection detection module helps figuring out the objects in the surroundings and theKeyboard helps in generating the natural language in the text form.

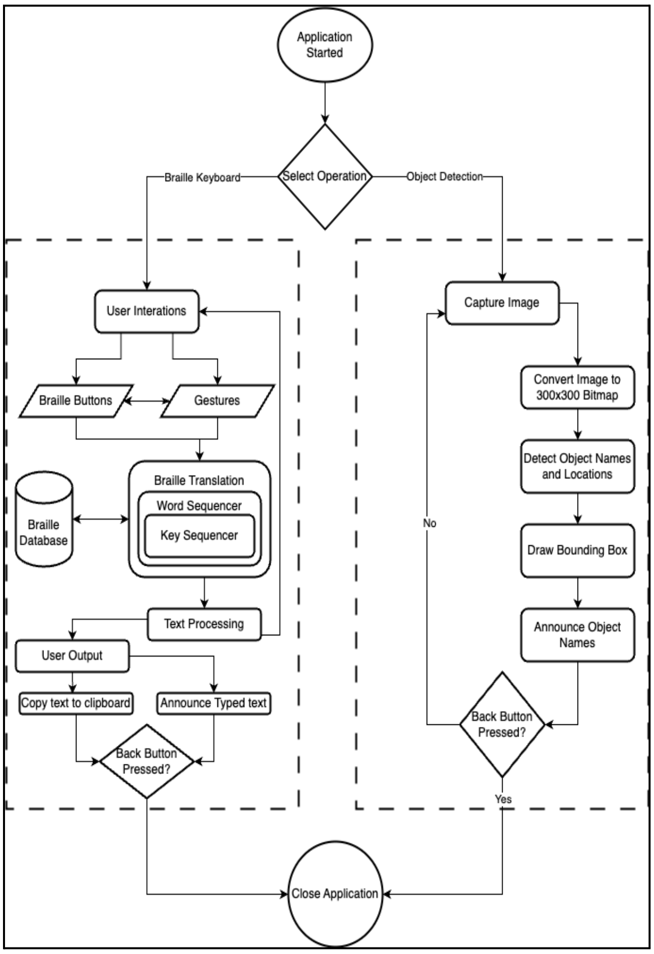

Figure 2. Control Flow of the Application Suite

3.1 Object Detection Module

The field of Computer Vision has witnessed dramatic improvements over the last decade [3] thanks to the availability of large scale data and computing resources. By teaching computers to learn and process from images and videos these techniques have found widespread acceptance in a wide range of domains including video surveillance [10], autonomous driving [11] and facial recognition [12]. Although the use of these technologies has been widespread, their reliance on the availability of large computational power makes their implementation on mobile phones a challenging task.

Given the constraints in which mobile applications operate and the requirements to seamlessly generate results in real time, it is essential to use an object detection technique that is simple and easy to integrate with different applications. The SSD: Single Shot Detector [13] emerged as the top contender to satisfy these requirements primarily because of their ability to generate effective predictions in real time. It requires only one scan of the image for performing object detection. SSD also addresses a key limitation of its predecessor YOLO [14] in identifying objects that are nearby which makes SSD more suitable for our application.

The algorithm starts by fitting one of a predefined set of bounding boxes at each location in the image and predicts the box which best captures an object. Once the bounding box has been predicted, the algorithm predicts the confidence which it has about the presence of each of the known objects inside the box. In order to make these predictions the algorithm uses a feed forward neural network architecture.

This network consists of a truncated version of a typical image classification model like VGG16 as its base network which is followed by progressively decreasing feature layers that use kernel functions to generate predictions about the position and category of the object. When an image is passed as an input the network generates predictions of the names of objects and their locations in the image. Each prediction is associated with a score that reflects the confidence which the model has in the prediction.

In this project, a MobileNet is used as a base network with SSD. MobileNet is another Convolutional Neural Network (CNN) which has been designed to achieve high performance to work on mobile phones and embedded devices[15]. This model is available for download on the Tensorflow website. It expects the input image to be in the form of a 300x300 tensor with 3 color channels. The model can predict 90 different objects typically encountered by a person in his daily life such as laptop, water bottle, apple, cell phone, etc. The predictions made by the model have been thresholded at 50% to ensure that the application gives reliable outputs.

3.2 Braille Keyboard Module

The Braille Keyboard enhances digital accessibility[4] for visually impaired individuals on smartphone users. It empowers them with a comprehensive experience[5], going beyond standard touch and speech commands[6][7] by integrating buttons and gestures for text generation.

The Braille Keyboard's specifications encompass several key elements for enhanced accessibility. It facilitates user interaction through Braille button tapping or predefined gestures on the dedicated gesture pad. This includes gesture recognition for seamless engagement and Braille translation that converts input from both button presses and gestures into readable text, accommodating contracted and uncontracted Braille. Moreover, it includes text processing capabilities for functions like speech generation, text copying, and message composition. The user output is presented via the mobile app's interface, either as audio feedback or in the clipboard.

In terms of design, the component architecture revolves around user interface (UI) elements such as the Braille Buttons, Gesture Pad, and Text Display, offering intuitive interaction and feedback[8]. The Gesture Recognition Module, smoothly implemented through Android Studio, enables various gestures like space, backspace, letter construction, text-to-speech, copying text, and word selection. Additionally[9], the Braille Translation Engine handles the conversion process by mapping Braille input to the appropriate characters based on the pressed buttons.

Overall, the Braille Keyboard strives to diminish the accessibility gap for visually impaired individuals in the digital sphere. It aims to achieve this by providing a versatile, intuitive, and inclusive mobile application experience across multiple platforms, enhancing the independence and engagement of visually impaired users.

4. IMPLEMENTATION

The application was developed in Android Studio Giraffe. XML based layouts were used for developing all user interfaces and the backend code has been developed using Kotlin and Java. The application consists of 3 activities: MainActivity, BrailleKeyboardActivity, and ObjectDetectionActivity. The MainActivity presents users with two options of either using the Braille keyboard or opening the object detection component.

As soon as control is passed to the ObjectDetectionActvity the phone's back camera will be started and new images get captured continuously. Operations related to the camera are handled by Camera2API to send and receive content. Once the ObjectDetectionActivity performs preprocessing on those images and passes the result to a SSDLite-MobileNet CNN model, which generates its predictions and sends them back to the ObjectDetectionActivity. The predictions are used to draw bounding boxes on the image and the names of detected objects are forwarded to the text-to-speech component.

As and when the control is passed to the BrailleKeyboardActivity, the buttonController and the GestureComponents are activated. The Button Controller has the components of Word Sequencer which is used to build words from the generated Key Sequencer. The generated intermediate is quired to the database, which gives out the english text which is then forwarded to the text-to-speech component.

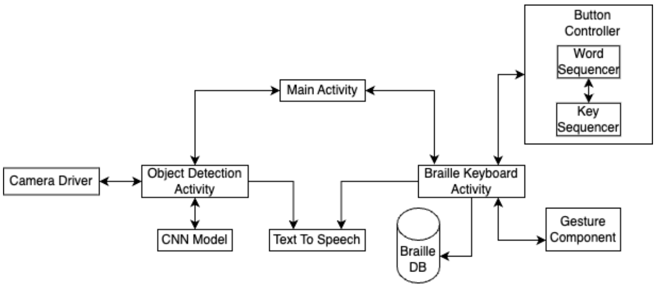

Figure 3. Application Components

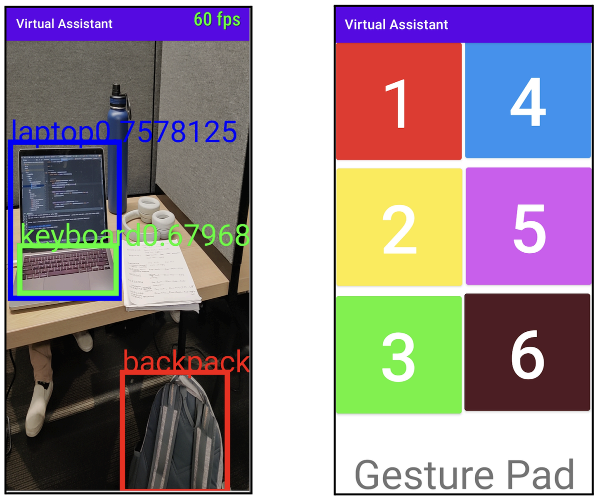

Figure 4. Application screenshots

5. DEMONSTRATION

The demonstration setup is designed to empower visually impaired individuals through a combination of cutting-edge technologies. The demonstration setup will include objects needed for detection, android smartphone and a visually impaired user. The visually impaired user chooses the functionality they would like to use- object detection or Braille Keyboard. If the user opts for object detection, a camera or sensors capture visual information, which is processed and interpreted on the connected Android smartphone. The system provides audio feedback or descriptions of detected objects, enhancing the user's awareness of their surroundings. Alternatively, if the user selects the Braille Keyboard, the smartphone integrates a virtual Braille input device, allowing the user to input text or navigate the device using Braille language.

6. CONCLUSION

In conclusion, our project stands as a successful response to the pressing needs of the visually impaired community in an increasingly digital landscape. Through the integration of AI and the innovative Braille keyboard for smartphones, we have developed a transformative solution that markedly elevates the daily experiences of individuals with visual impairments. Our application seamlessly combines real-time object detection, an efficient text-to-speech converter, and a Braille keyboard, not only fostering greater independence but also enhancing the overall quality of life for users. By effectively addressing challenges in object identification, simplifying smartphone typing, and supporting daily tasks, our application exemplifies a commitment to leveraging technology for social good, embodying the core principles of innovation and empathy.

Beyond its immediate impact, our project holds substantial potential in the realm of assistive technology, opening new avenues for the visually impaired to interact with their surroundings and serving as a benchmark for future technological advancements in this critical area. Moving forward, our focus remains on refining our mobile application, incorporating user feedback, and exploring additional features to continually better serve and empower the visually impaired community.

7. REFERENCES

[1] "Vision Impairment and Blindness." World Health Organization, World Health Organization, www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment

[2] Marson, Stephen & Harrington, Charles & Walls, Adam. (2013). Teaching introductory statistics to blind students. Teaching Statistics. 35. 10.1111/j.1467-9639.2012.00510.x.

[3] Ponte, H. (2020). *A New Decade of Computer Vision*. [online] Medium. Available at: https://towardsdatascience.com/a-new-decade-of-computer-vision-ebc98712e95b

[4] Šepić, Barbara, Abdurrahman Ghanem, and Stephan Vogel. "BrailleEasy: one-handed braille keyboard for smartphones." Assistive Technology. IOS Press, 2015. 1030-1035.

[5] Ellis, Kirsten, et al. "Bespoke Reflections: Creating a one-handed braille keyboard." Proceedings of the 22nd International ACM SIGACCESS Conference on Computers and Accessibility. 2020.

[6] Church, Alex, et al. "Deep reinforcement learning for tactile robotics: Learning to type on a braille keyboard." IEEE Robotics and Automation Letters 5.4 (2020): 6145-6152.

[7] Alnfiai, Mrim, and Srini Sampali. "An evaluation of the brailleenter keyboard: An input method based on braille patterns for touchscreen devices." 2017 international conference on computer and applications (ICCA). IEEE, 2017.

[8] Eom, Tae-Jung, Jung-Bae Lee, and Byung-Gyu Kim. "Design of Smart Phone-Based Braille Keyboard System for Visually Impaired People." Convergence Security Journal 12.1 (2012): 63-70.

[9] Romero, Mario, et al. "BrailleTouch: designing a mobile eyes-free soft keyboard." Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services. 2011

[10] P. K. Mishra and G. P. Saroha, "A study on video surveillance system for object detection and tracking," 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 2016, pp. 221-226.

[11] Gupta, A., Anpalagan, A., Guan, L. and Khwaja, A.S. (2021). Deep learning for object detection and scene perception in self-driving cars: Survey, challenges, and open issues. *Array*, 10, p.100057. doi:https://doi.org/10.1016/j.array.2021.100057.

[12] Griffin, J.W. and Motta-Mena, N.V. (2021). Face and Object Recognition. *Encyclopedia of Evolutionary Psychological Science*, pp.2876–2883. doi:https://doi.org/10.1007/978-3-319-19650-3_2762.

[13] Liu, W. *et al.* (2016). SSD: Single Shot MultiBox Detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds) Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science(), vol 9905. Springer,Cham.https://doi.org/10.1007/978-3-319-46448-0_2

[14] Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. (2015). *You Only Look Once: Unified, Real-Time Object Detection*. [online] arXiv.org. Available at: https://arxiv.org/abs/1506.02640.

[15] Howard, A., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T. and Andreetto, M. (2017). *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*. [online] Available at: https://arxiv.org/pdf/1704.04861.pdf.