2022 International Conference on Electrical, Computer and Energy Technologies (ICECET) | 978-1-6654-7087-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICECET55527.2022.9872538

Autonomous Classification and Spatial Location of Objects from Stereoscopic Image Sequences for the Visually Impaired.

Themba M Sivate
Department of Electronic and Computer Engineering
Durban University of Technology
Durban, Kwa-Zulu Natal,
Republic of South Africa
sivate2@gmail.com

Nelendran Pillay
Department of Electronic and Computer Engineering
Durban University of Technology
Durban, Kwa-Zulu Natal,
Republic of South Africa
trevorpi@dut.ac.za

Kevin Moorgas
Department of Electronic and Computer Engineering
Durban University of Technology
Durban, Kwa-Zulu Natal,
Republic of South Africa
kevinm@dut.ac.za

Navin Singh
Department of Electronic and Computer Engineering
Durban University of Technology
Durban, Kwa-Zulu Natal,
Republic of South Africa
navins@dut.ac.za

***Abstract* - One of the main problems faced by visually impaired individuals is the inability or difficulty to identify objects. A visually impaired person usually wears glasses that help to enlarge or focus on nearby objects, and therefore heavily relies on physical touch to identify an object. There are challenges when walking on the road or navigating to a specific location since the vision is lost or reduced thereby increasing the risk of an accident. This paper proposes a simple portable machine vision system for assisting the visually impaired by providing auditory feedback of nearby objects in real-time. The proposed system consists of three main hardware components consisting of a single board computer, a wireless camera, and an earpiece module. YOLACT object detection library was used to detect objects from the captured image. The objects are converted to an audio signal using the Festival Speech Synthesis System. Experimental results show that the system is efficient and capable of providing audio feedback of detected objects to the visually impaired person in real-time.**

Keywords—visually impaired, single-board computer, portable, Bluetooth, computer vision, audio feedback, object detection

I. INTRODUCTION

Globally, at least 2.2 billion people have a near or distance vision impairment [1]. Visually impaired persons showed a higher level of anxiety, depression, and emotional distress [2]. A visually impaired individual needs to be able to navigate places, but that may be challenging if you cannot see the objects surrounding you. Some technologies were developed in the past to assist visually impaired individuals to navigate places with ease. One of the most common technologies is the walking stick equipped with sensors to detect nearby objects. The major drawbacks of the walking stick are the short distance on object detection and that the user always needs to carry it by hand and manually search for nearby obstacles. There are several proposed solutions to the challenges faced by visually impaired personnel by incorporating wearable sensors. In [3], the authors proposed an intelligent walking stick for blind safety protection. The walking motion can be monitored and controlled using a short-range ad hoc network, where required safety conditions can be specified. The main drawback of this method is that it only operates for short-distance object detection and cannot identify objects that are outside the device range. In [4], a machine vision system using 3D audio sensory feedback to assist with navigation for the visually impaired was proposed. Two Raspberry Pi microcomputers were used in conjunction with a standalone remote server responsible for performing the object detection. The final prototype consisted of a battery-powered headset coupled with wired earphones. While the system proved to be useful and accurate for navigation, when tested in a real-life environment it consisted of a bulky headset and suffered from higher power supply requirements.

A cognitive navigation IoT system for the visually impaired was proposed by [5]. The system however used wired connections and could not operate under low lighting conditions. In [6], the authors proposed a system equipped with four ultrasonic sensors and one infrared sensor. All distance measuring transducers were connected to a single microcontroller, but additional hardware was required for processing the preceding terrain surfaces. The additional equipment and multiple sensors that were used caused considerable weight increases to the portable arrangement. In [7], a vision system to recognize only pure colors was proposed. This system identified different color bands using a TCS230 color sensor and an Arduino Microcontroller board. The system had been successfully deployed in real-time by both blindfolded and blind individuals under indoor and outdoor conditions. However, the system was trained to recognize only a limited number of colors and was not expanded to object detection. In this study, we propose a low-cost modular and portable computer vision system to solve the computer vision problem and minimize the weight. The proposed system makes use of a single-board computer (SBC), wireless camera, and an earpiece. The modular system takes advantage of wireless data transmission protocols (Wi-Fi and Bluetooth), in which data processing can be achieved in real-time using low-cost computer hardware located possibly inside a pocket or a handbag. The SBC processes the received image and uses a speech synthesis program to convert the identified

978-1-6654-7087-2/22/$31.00 ©2022 IEEE

object into an audio signal of the corresponding object which is then played back on a Bluetooth-enabled earpiece.

This paper is organized as follows: Section II discusses the proposed hardware architecture of the system. Section III describes the methodology for image detection and speech processing. Section IV reviews the experimental results obtained from the proposed system. Lastly, Section V will conclude the paper and provide some recommendations for future work.

II. System Hardware Architecture

The proposed system architecture is shown in Fig 1. The SBC selected for this project is the 'ROCK64' produced by Pine64®. It is a 4GB RAM board powered by a Rockchip RK3328 and controlled by the Debian Linux operating system. The board is not equipped with onboard Bluetooth and WIFI interfaces, however, it can be easily connected to WIFI and Bluetooth USB adaptors. The Bluetooth earpiece selected for this arrangement is the Lenovo® HX106 which has 120 hours on standby, 300 minutes when on call, and 20 hours when playing back audio.

The portable wireless camera selected for this project is SJCAM® C100+ with continuous operational use of 150 minutes. The wireless camera streams live captured images to the SBC and are unidirectional. In addition, the operation between the SBC and the earpiece is also unidirectional. However, under normal circumstances, a typical commercial Bluetooth earpiece can also send commands to the main hardware device to control the media player, answer, or initiate a call, but these commands do not apply in this study. The SBC is powered by a 10000mAh battery rated at 5V, 2.1A output. Both wireless devices are powered individually by internal rechargeable batteries. Further, to increase the battery life of the system, the software places the hardware devices on standby mode when processing is not required. The system automatically shuts down when the power supply voltage is low. The overall weight of the proposed setup is 400 grams.

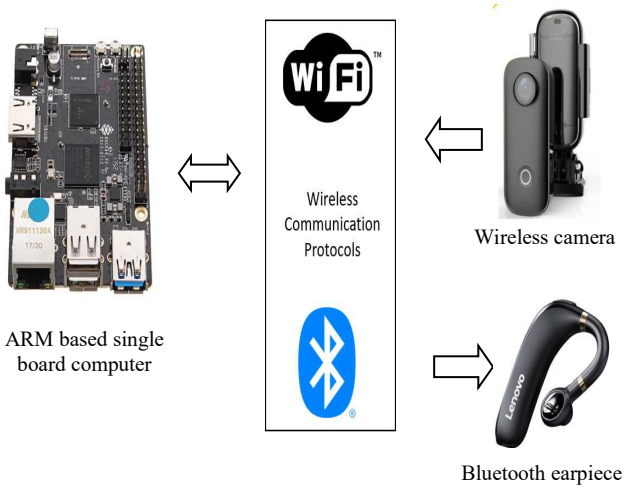


Fig. 1. Block diagram of the autonomous image classification system.

III. Proposed Methodology

A. System operation.

The system operational sequence is shown in Fig. 2. OpenCV library was used to capture the frames or sequence of images in real-time. OpenCV is an open-source library that includes several hundreds of computer vision algorithms [8]. The main benefit of OpenCV is its relative ease of use and flexibility. The data is then sent to the SBC for processing. YOLACT [9] image classification library was then used with a pre-trained model to classify data in the captured image frame. Machine learning classification tools in the library package provide an effective way to achieve object detection in real-time.

The system autonomously executes commands to capture and process images, output audio feedback and control overwrite sequences. Namely, the system captures the image, stores it on board a 16 GB memory storage device, processes the image to identify objects of interest, and outputs the detected object to a corresponding audio file. Speech synthesis is achieved through the Festival Speech Synthesis C++ library which is open source [10]. Previous images are overwritten. The system continues to loop in this sequence with one operating log file. Logs can be generated if required for post-analysis of data interrogation. If log files are generated, a threshold of the log file size needs to be specified to prevent excessive memory usage.

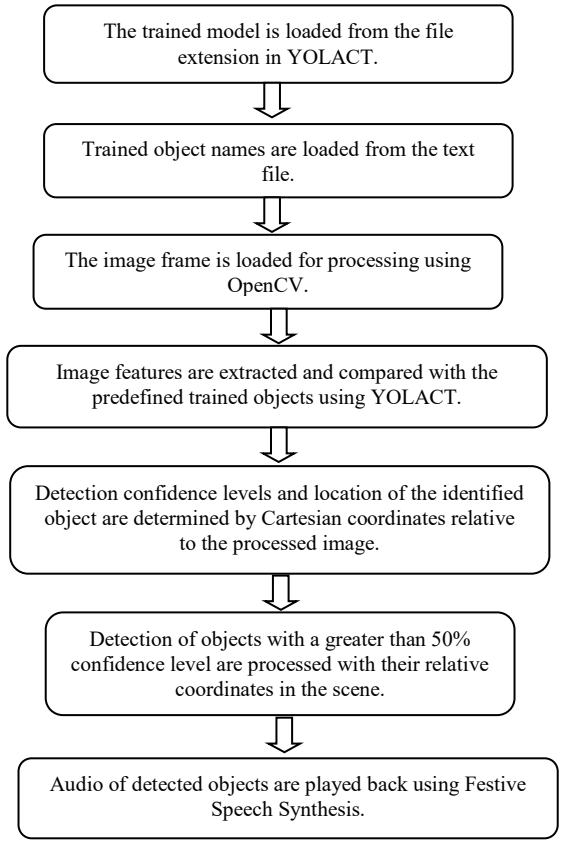


Fig. 2. System operation sequence.

B. *Object detection algorithm using YOLACT*

In this work, YOLACT was used to detect objects in the captured images. YOLACT is based on the ResNet-101 convolutional neural network algorithm and FPN (Feature Pyramid Networks) [11]. YOLACT is a real-time instance segmentation algorithm that divides the instance segmentation task into two parallel branches, namely 'Protonet' and 'Prediction'. Protonet acts as a semantic segmentation model [12], which is the process of grouping parts of the image which belong to the same object class. Fig. 3 illustrates the process of semantic segmentation. The prediction branch is used for forecasting per-instance mask coefficients [13]. Namely, YOLACT predicts a set of linear combination coefficients per instance and produces instance masks by linearly combining the prototypes with the mask coefficients [13]. Fig. 4 shows the components and steps used by YOLACT for image prediction and classification. Object detection using the YOLACT algorithm is accomplished using a pre-trained model capable of detecting predefined objects in a particular scenario. In this work, the system was built to detect objects found in a typical street suburb. The algorithm was trained to generate a model that can be used to detect common objects that are usually located in a suburban environment. However, to evaluate if the system can handle image processing without misidentifying real-world objects, the model was also trained to include objects that are not normally encountered within this environment, namely common household objects. An unknown object is defined if the confidence level is less than or equal to 10%. Table I shows the sample of common objects that YOLACT was trained to detect.

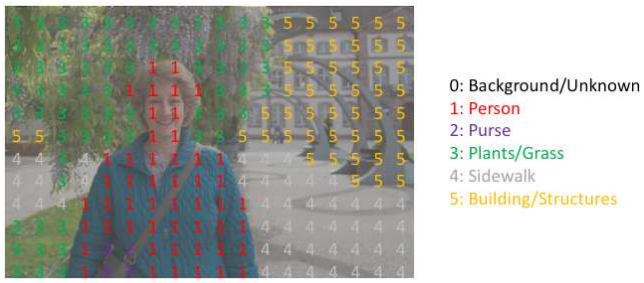


Fig. 3. Semantic segmentation of an input image

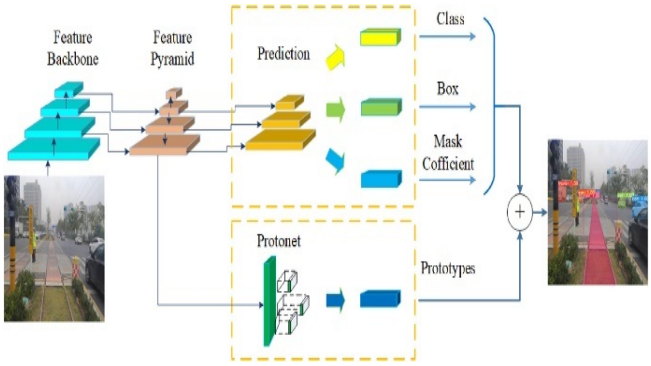


Fig. 4. YOLACT prediction diagram

TABLE I. SAMPLE OF COMMON OBJECTS

List of objects to be trained	
traffic light	laptop
person	keyboard
car	mouse
bus	tv
motorcycle	oven
fire hydrant	bed

C. *Speech processing system using Festival Speech Synthesis*

To convert the detected object into audio, a speech synthesis system is used. The Festival Speech Synthesis [10] was chosen for the proposed system since it is a well-established text-to-speech engine that supports the Linux environment. Festival Speech Synthesis originally developed by the Center of Speech Technology at the University of Edinburgh provides all the modules necessary for text-to-speech conversion, along with an environment suitable for researching all areas of speech synthesis [14]. Festival offers a general framework for building speech analysis computation and offers full text-to-speech conversion through an Application Programming Interface (API) [15]. The APIs range from shell level, C++ library, JAVA library, and a graphical user interface. Festival is modular and uses a simple framework, commonly known as 'blackboard architecture' which is centered on a common data structure called the 'utterance', which is transferred between modules within the system [16]. The core architecture of Festival is written in C++, while the modules on the other hand can be written in C++ or other higher-level languages [16]. Since Festival was built in C++, it makes it compatible with the proposed system, however, a shell API was used and called within the C++ application. The process simplifies the development and makes it possible to convert text to audio playback with a single line of code without including any header files which reduces execution time.

IV. EXPERIMENTAL RESULTS

A. *Detection of objects using the proposed methodology*

Fig. 5 (a) and Fig.5 (b) show the results obtained from the object detection algorithm of the proposed system. Objects of interest were isolated based on the items listed in Table 1 in an outdoor environment. Identified objects with a greater than 50% confidence level threshold are outputted to an audible sound signal relative to their given position. In Fig. 5a, several cars were detected of which four had high confidence percentages of 97.4%, 66.0%, 97.7%, and 98.4%. The audio playback was '*car*' for the detected objects in the scene. In Fig. 5b, YOLACT was able to correctly identify one object as a '*person'* with a correct boundary at a confidence level of 17.3% and several vehicles as '*car*' with confidence levels ranging from 26.7% to 99.4%. The audio output was however given as '*car*' for the image since the '*person'* was below the 50% threshold.

(a) Processed street image

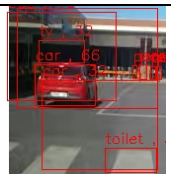
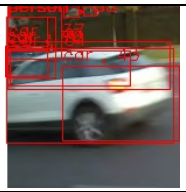
(b) Processed parking lot image

Fig. 5. Object detection results of the proposed system using YOLACT.

B. *Comparison of real-world scenes with acceptable image quality*

To test the efficacy of the object detection algorithm, YOLACT was compared with TensorFlow [17], which is another popular object detection library. Table II shows the comparison results between YOLACT and TensorFlow for several captured scenes typically found in a metropolitan urban environment. In this study, both applications were programmed to process all possible objects detected in the scene, irrespective of their associated confidence percentage level. It was observed that TensorFlow was relatively fast to process each scene but at the expense of lower object detection confidence percentage levels. In contrast, YOLACT resulted in an improved object detection reliability but exhibited significant time delays.

TABLE II. YOLACT VERSUS TENSORFLOW FOR TYPICAL URBAN ENVIRONMENT

Captured images		***Object of interest***	***Confidence level (%)***		***Delay time (sec)***	
YOLACT object detection	***TensorFlow object detection***		***YOLACT***	***TensorFlow***	***YOLACT***	***TensorFlow***
		Person	99.4	73.0	5.071	0.181
		Car	95.4	66.0	5.791	0.095
		Vehicle in motion	63.3	43.0	4.99	0.095
		Bus	90.4	55.0	6.586	0.101
		Traffic light	87.8	64.0	7.549	0.113
		Complex street scene	87.8	64	7.549	0.113

Furthermore, it was also observed that the YOLACT algorithm displayed improved accuracy when determining boundary coordinates for the subsequent images. In some instances, TensorFlow identified objects that did not exist in the real-world scene and occasionally provided a bounded area that was outside the detected object of interest.

C. *Comparison of real-world scenes with blurry image quality*

To evaluate the efficacy of the proposed methodology, the system was subjected to blurry images. Fig. 6 shows a real-world footpath image whereby clear and blurry objects were processed. This can be rectified by using image stability correction software at the expense of increased image processing computational delays.

Blurry images can also be reduced by considering a camera with built-in image stabilizing optics. Obtaining a blurry image may be due to little or no image stabilization correction. It was observed that blurry images reduced the confidence levels of the identified objects in the scene.

Using a confidence threshold of 50%, in Fig. 6(a) YOLACT was able to identify a 'bus' with an 88.7% confidence level while the blurry image resulted in the same object being identified as a 'car' with 51.1% accuracy. Fig. 6(b) provides the object detection results using Tensorflow. Although the bus was identified as a 'truck' in the clear image with 59% accuracy, the blurry image resulted in an incorrectly defined object with 56% confidence.

(a) YOLACT image object detection

(b) Tensorflow image object detection

Fig. 6. Comparison of a clear image versus blurry image using YOLACT and Tensorflow processing

D. Feature extraction latency

Several tests were conducted to evaluate the system time delays of observed scenes under different light conditions. Table III shows the comparative delay time between scenes with low light conditions and well-lit scenes. It also shows the number of detected objects, the resulting image output file size, and output latency. For the observed images, the well-lit scene the image processing software was able to detect 6 objects and had a delay time of 10.18 seconds. In contrast, in a poorly lit environment, the time delay was 9.63 seconds even though 8 objects were detected.

TABLE III. PROCESSED DATA FOR DIFFERENT LIGHT CONDITIONS

Ambient light condition	*Number of objects detected*	*Output image size (kilobytes)*	*Latency (sec)*
Well-lit environment	6	1400	10.19
Poorly-lit environment	8	914	9.63

V. CONCLUSION AND RECOMMENDATIONS

In this study, a system was proposed to assist the visually impaired to navigate an outdoor environment with the aid of an autonomous machine vision system that provides an audible signal relative to detected objects in front of the person. Preliminary results obtained from the prototype demonstrate that it is possible to implement a low-cost computer vision system that is portable, modular, non-obstructive, and reliable. Owing to the modular nature of the hardware architecture, it was possible to produce a relatively lightweight system that could be mounted on the body of a visually impaired person. Physical hardware connections between the modular devices are eliminated by means of wireless data communication protocols. The hardware architecture can be improved by incorporating the camera with a custom-built printed circuit board to further reduce the physical footprint and weight of the system. The YOLACT image object identification software resulted in superior results when compared to Tensorflow under different captured scenes and image quality albeit at increased time delays. To reduce the time delay, it is recommended to ignore objects which are deemed to be located far away from the visually impaired individual. Furthermore, increased processing performance may be achieved by disregarding objects that have very low confidence levels. Further study is required to determine the tradeoff benefits between low image resolution processing speed versus the accuracy of the object detection.

REFERENCES

[1] World Health Organization. (2021) "Blindness and vision impairment". [eBook] Available at:https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment.

[2] R. Shu, C. Liu, H. Liang and Y. Liang, "Potential mediators of the relationship between vision impairment and self-rated health in older adults: A comparison between long-term care insurance claimants in residential care institutions versus those living in the community", Geriatric Nursing, vol.44, 2022, pp.259-265

[3] R. Khlaikhayai, C. Pavaganun, B. Mangalabruks and P. Yupapin, "An Intelligent Walking Stick for Elderly and Blind Safety Protection", Procedia Engineering, vol. 8, 2011, pp.313-316

[4] G.S. Aakash Krishna, V.N. Pon, S. Rai and A. Baskar, "Vision System with 3D Audio Feedback to assist Navigation for Visually Impaired", Procedia Computer Science, vol. 167, 2020, pp.235-243.

[5] C.P.G. Mallikarjuna , R. Hajare and P.S.S. Pavan, "Cognitive IoT System for visually impaired: Machine Learning Approach", Materials Today: Proceedings, vol. 49, 2022, pp.529-535.

[6] S. Sirouspour, E. Hossain, R. Khan, R. Muhida and A Ali, "Analysis and Implementation for a Walking Support System for Visually Impaired People", International Journal of Intelligent Mechatronics and Robotics, vol. 1, 2011, pp.46-62.

[7] M.D.C. Perera, M.I.M. Amjath, H.M.S.C.R. Heenkenda and V. Senthooran, "Assist System for Visually Impaired People to Recognize Pure Colours", Trincomalee International Conference, 2017.

[8] OpenCV: Introduction. [Electronic resource]. URL: https://docs.opencv.org/4.x/d1/dfb/intro.html

[9] D. Bolya, C. Zhou, F. Xiao and Y. J. Lee, "YOLACT: Real-Time Instance Segmentation," 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 9156-9165

[10] P. Taylor, R. Caley and H. Zen “The Festival Speech Synthesis System”, 1997. Available at: https://www.cstr.ed.ac.uk/projects/festival/.

[11] Z. Shang, X. Wang, Y. Jiang, Z. Li and J. Ning, "Identifying rumen protozoa in microscopic images of ruminant with improved YOLACT instance segmentation", Biosystems Engineering, vol. 215, 2022, pp.156-169

[12] Z.Zhang, S. Huang, X. Liu, B. Zhang and D. Dong, "Adversarial attacks on YOLACT instance segmentation", Computers & Security, vol. 116, 2022

[13] J. Jordan, “An overview of semantic image segmentation”, 2018. Available at: https://www.jeremyjordan.me/semantic-segmentation.

[14] R.A.J. Clark, K. Richmond and S. King, “Multisyn: Open-domain unit selection for the Festival speech synthesis system”, Speech Communication, vol. 49, 2007, pp.317-330

[15] R.A.J. Clark, K. Richmond and S. King, “Festival 2 - Build your own general purpose unit selection speech synthesiser”, 5th ISCA Speech Synthesis Workshop, 2004, pp.173-178

[16] P. Taylor, A. Black and R. Caley, “The architecture of the Festival speech synthesis system”, The Third ESCA Workshop in Speech Synthesis, 1998, pp. 147–151

[17] P.S. Janardhanan, "Project repositories for machine learning with TensorFlow", Procedia Computer Science, vol. 171, 2020, pp.188-196