Original Article

Visual Cues to Voice Cues

J. Thamil Selvi[1], S. Keerthana Devi[2], S. Shreya[3], M. Sanjana[4]

[1]Professor, Department of Electronics and Communication Engineering, Sri Sairam Engineering College, Chennai, India.

[2]IV Year UG Student, Department of Electronics and Communication Engineering, Sri Sairam Engineering College, Chennai, India.

[3]IV Year UG Student, Department of Electronics and Communication Engineering, Sri Sairam Engineering College, Chennai, India.

[4]IV Year UG Student, Department of Electronics and Communication Engineering, Sri Sairam Engineering College, Chennai, India.

[1]thamilselvi.ece@sairam.edu.in, [2]e8ec168@sairamtap.edu.in, [3]e8ec131@sairamtap.edu.in, [4]e8ec129@sairamtap.edu.in

Abstract

The project is aimed at developing an object identifier for visually challenged. The proposed work uses ESP32CAM which acts as human eye. When visually challenged people navigate in known or unknown, indoor or outdoor environments, as they face troubles due to inaccessible infrastructure and social challenges, which in turn improve quality of Visually Impaired through visual information to the users. The idea is to capture real-time images of objects through ESP32CAM. Using the machine learning technique, object detection is done. The captured image is analyzed and converted into text. Further, the rule of text-to-speech is used, the text is then converted into sound using the Python module. The output of Text-to-speech is then amplified using an audio amplifier which is then heard through an earphone.

Keywords: Object Detection, ESP32 Camera, Machine Learning Algorithm, Computer Vision, Video Recognition.

DOI: 10.47750/pnr.2022.13.S03.062

INTRODUCTION

Blindness is a very common disability which cannot be corrected with glasses or contact lenses. Eyes are a precious gift to mankind. Health Organization (WHO), reports 285 million people are visually impaired worldwide. Among them 90 percent of the visually impaired people live in developing countries. Vision allows us to see the world around us, maintain our balance and also helps to navigate and aid to live normal life [1]. The human eye works like a camera. It is very essential for blind people to recognise the objects around them, and to use them in day-to-day life. To overcome the disability of blindness, various traveling support systems are developed, to obtain information of surroundings more efficiently [2]. Literature reports many android application and mobile phones to aid blind people to lead normal life [3].

Many papers deal with the Android based smart cane with sensors for an obstacle detection and guide them towards their destination [4]. A visually impaired person requires assistance for everyday chores. To guide visually impaired people similar to smart cane, smart gloves have been introduced [5]. Computer vision is emerging as the advanced technology. Computer vision is used for object detection but very challenging on images due to its contrast and intrinsic parameters. [6]. An object tracker identifies an object within multiple frames and assigns labels to each object. YOLO is an efficient tool attempted by author which uses single step object detection and classification method [7]. YOLO is a new approach to object detection. YOLO is used to train full images and optimizes directly optimizes [8], and achieves more precision compared to other real-time systems. The ESP32-CAM captures video and the object detection module [9] is implemented.

In this work, we have attempted to detect object using machine learning techniques for visually impaired people. The objects which are being captured are first identified and detected and converted to text. The text will then be converted to a voice. The end result which is a voice will be heard through an earphone by the Visually challenged people.

METHODOLOGY

A. Image Database

For Object detection purpose, online database trained with multiple images have been used. Microsoft have published dataset called The Microsoft Common Objects in Context (MS COCO) dataset and a trained a CNN to detect objects in large-scale and captioning dataset and it is widely used by Machine Learning and Computer Vision engineers for

various computer vision projects [10]. The primary goal of computer vision is understanding the visual scenes, it involves recognizing what objects are present, determining the object's attributes and characterizing it.

- Input images from COCO Dataset is 121,408 images
- Annotated images from COCO Dataset is 883,331
- 80 classes of images using the COCO Dataset
- The COCO Dataset median image ratio is 640 x 480

The COCO-Stuff dataset of 91 stuff classes comes with pixel-wise annotations for a rich and diverse set [11].

B. Proposed Approach

The proposed model aims to address the visually impaired population. An attempt is made to design an intelligent assistive device which identifies the real world object and converted to speech and fed to visually challenged people. This in turn helps visually impaired people to lead independent and normal day to day life. The system architecture is shown in Fig. 1. It consists of ESP32 Camera, object detection, text-to-speech conversion.

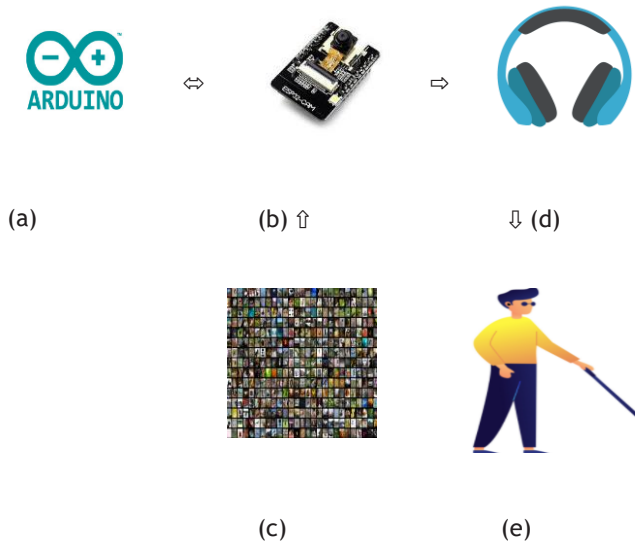


Fig. 1; (a) Arduino IDE (b) ESP32CAM (c) Trained Dataset (d) Headphone (e) Blind Person

ESP32-CAM

The ESP32 is one of the most popular wireless microcontroller boards. The ESP32 CAM is small in size, consumes low power camera module which is based on ESP32. Besides the OV2640 camera with a microSD card slot is also used to store and process the captured images. The ESP32 CAM is Interfaced FTDI programmer to upload code with the help of Arduino IDE. It can capture raw images with the maximum resolution of 1600x1200. Objects are captured using ESP32 CAM for detecting it. Fig. 2 shows how an ESP32 CAM can be connected to a USB connector FTDI programmer for programming it.

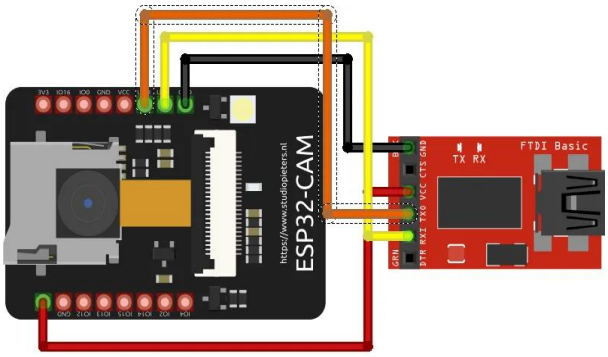


Fig. 2; Connection Diagram of ESP32CAM with FTDI Programmer

Object Detection

Object detection combines classification and localisation of objects in images and videos.

A bounding box is drawn around each object of interest in the image and assigns a class label. Every object class has its unique features that helps in classifying the objects. This feature is incorporated in the project to help blind subjects in identifying the objects in front of them. This in turn help them to get a sense of their surroundings.

Text to Speech

To convert text to speech Python library is used. The final output which is obtained as voice will be conveyed to the visually impaired through earphones to identify the objects in their surroundings.

C. Algorithm

In this proposed work, OpenCv and the pre-trained deep learning model i.e, YOLOv3 is used for the object detection.YOLOv3 is a real-time object detection algorithm that identifies specific objects in videos, or images even in live streaming video. YOLO uses features read by a deep convolutional neural network to detect an object [12]. The process of YOLOv3 is shown in Fig. 3.

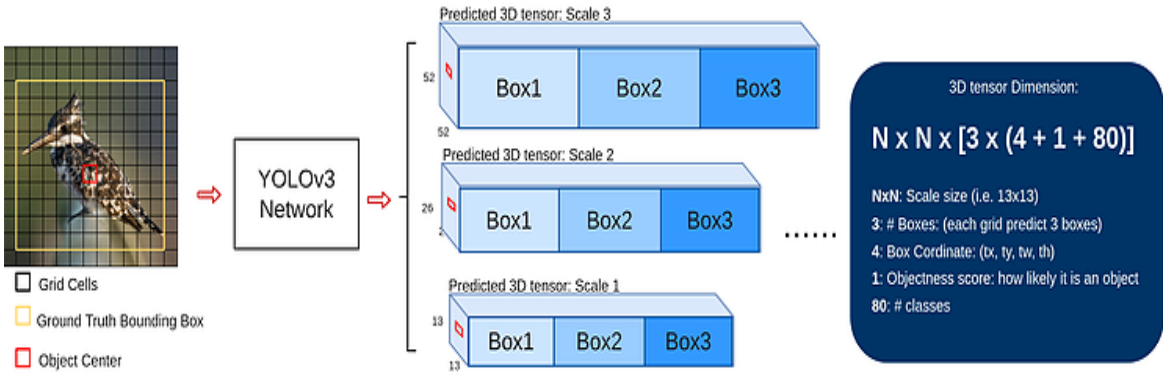


Fig. 3; Process Behind Machine Learning Algorithm

YOLO V3 is the convolution neural network with twenty four convolutional layers and two fully connected layers [13]. The first 4 elements represent the center at x,y, width and height. CNNs are classifier-based frameworks that interacts with input pictures as data arrays and aims to recognize patterns between them. YOLO provides equal weight to the classification and localization and find sum of squared error loss function [14]. The loss function defined as in yolo as.

$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$
$$+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$
$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2$$
$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2$$
$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2 \quad . (1)$$

Where,

$\mathbb{1}_{i}^{\text{obj}}$ denotes presence of an object in cell i.

$\mathbb{1}_{ij}^{\text{obj}}$ denotes a j th bounding box for object prediction in the cell i.

λ_{coord} and are regularization parametres.

In the Eqn. (1) localization mean-squared error is represented in first two parts and other parts represent classification error

Yolo with a single Layer neural network is applied to an image. The images are divided into SXS grids and based on the confidence score the bounding box are constructed and class label are assigned to individual object. Confidence score is the measures of accuracy of object present in the bounding box. tell The probability of object presence in the cell is measured using Class conditional probability.[15]

YOLO is one of the fastest object detection algorithms (forty-five frames per second) as compared to the R-CNN family (R-CNN, Fast R-CNN, Faster R-CNN, etc). Fig. 4 shows how the input image is applied with S X S grid in accordance with the usage of algorithm, furtherly the bounding boxes along with confidence are marked on image and with the class probability map, the object detected displayed.

The ratio of true predictions to the total number of input samples is termed as Accuracy. It can be using the following formula.

Accuracy = Correct predictions / All predictions

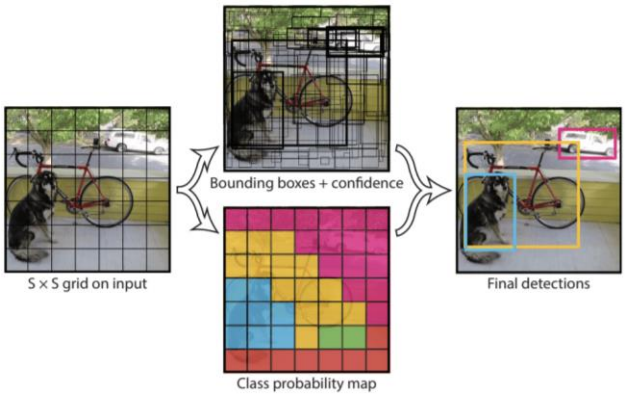


Fig. 4; The Process of Yolo Algorithm Applying on Image and its Final Detected Output

Precision is the ratio of number of True positive samples and total number of positive samples predicted in a certain class. It's formula is given by,

Precision =True positives/(True positives+False positives)

Recall is the ratio of examples which were predicted to belong to a true positive class with respect to all the samples that belong in that class and the can be used by the following formula.

Recall=True positives/(True positives+False negatives)

Result and Discussions

The visually impaired people always have the need to rely on others and to lead a dependent life. So, our work is mainly

focussed on making their lives easier by giving them the aid which would help them to identify objects around them. The implementation of computer vision machine learning algorithm is incorporated to detect objects and it also explains how convolution neural networks are trained on dataset that can detect objects.

For this study trained database of about 80 datasets using COCO datasets are used and tested. Where OpenCV 2.4 library is used to support machine learning method and Standard dataset has been considered for the result. The COCO dataset contains eighty classes, eighty thousand images and forty thousand validation images. In this regard, using an ESP32 camera, the following objects are captured from real time video streaming.

In this method of object detection model the objects captured are converted to text and then to voice for the visually impaired people. Detection of objects in a room using live streaming video frames is done with YOLO. The proposed method identifies potential areas using bounding boxes. The more overlapping bounding box of the captured images represent true region of interest in the network. A trained model attempts to identify the object in bounding box. The conventional object detection method is time consuming. The traditional method looks every part of image multiple times and classify the object. YOLO is unique technique to classify the object by looking the object only once. OpenCV is used to display the Results. The accuracy of object detected using YOLO is measured using the confusion matrix.

Fig. 5 shows the ESP32 camera integrated with the OV2640 camera.

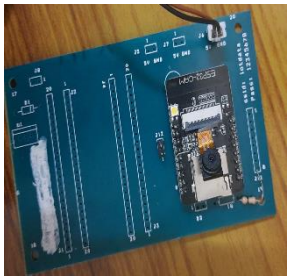


Fig. 5; ESP32CAM

The Arduino IDE is used for running the code for the operation of the ESP32 camera. The objects which are captured in the live video would be compared with the trained database of COCO using machine learning algorithm and hence the output is obtained where the objects are surrounded by a bounding box with labels. Fig. 6 represents the trained dataset in the COCO database obtained from our process and Fig. 7 represents the detected objects array are numbered whereas the non-detected objects array are empty.

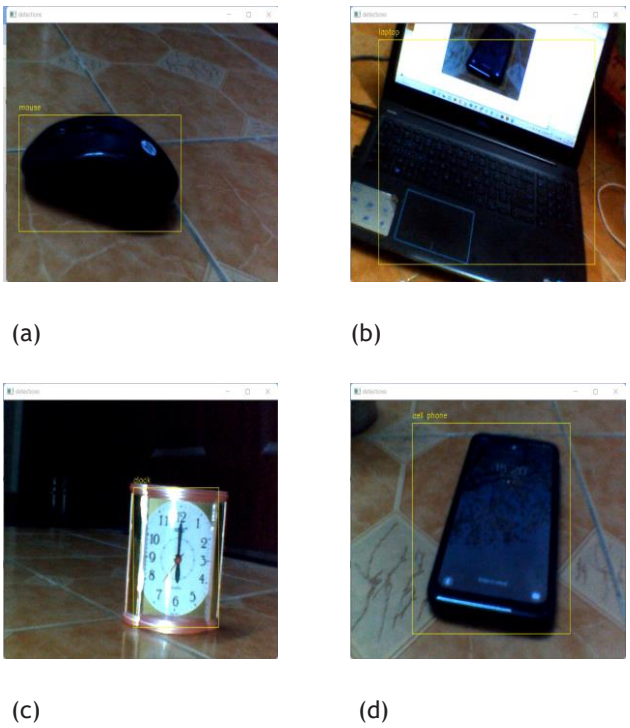
{'person': (0, 255, 255), 'bicycle': (0, 255, 255), 'car': (0, 255, 255), 'motorbike': (0, 255, 255), 'aeroplane': (0, 255, 255), 'bus': (0, 25
5, 255), 'train': (0, 255, 255), 'truck': (0, 255, 255), 'boat': (0, 255, 255), 'traffic light': (0, 255, 255), 'fire hydrant': (0, 255, 255),
'stop sign': (0, 255, 255), 'parking meter': (0, 255, 255), 'bench': (0, 255, 255), 'bird': (0, 255, 255), 'cat': (0, 255, 255), 'dog': (0, 25
5, 255), 'horse': (0, 255, 255), 'sheep': (0, 255, 255), 'cow': (0, 255, 255), 'elephant': (0, 255, 255), 'bear': (0, 255, 255), 'zebra': (0, 2
55, 255), 'giraffe': (0, 255, 255), 'backpack': (0, 255, 255), 'umbrella': (0, 255, 255), 'handbag': (0, 255, 255), 'tie': (0, 255, 255), 'suit
case': (0, 255, 255), 'frisbee': (0, 255, 255), 'skis': (0, 255, 255), 'snowboard': (0, 255, 255), 'sports ball': (0, 255, 255), 'kite': (0, 25
5, 255), 'baseball bat': (0, 255, 255), 'baseball glove': (0, 255, 255), 'skateboard': (0, 255, 255), 'surfboard': (0, 255, 255), 'tennis racke
t': (0, 255, 255), 'bottle': (0, 255, 255), 'wine glass': (0, 255, 255), 'cup': (0, 255, 255), 'fork': (0, 255, 255), 'knife': (0, 255, 255),
'spoon': (0, 255, 255), 'bowl': (0, 255, 255), 'banana': (0, 255, 255), 'apple': (0, 255, 255), 'sandwich': (0, 255, 255), 'orange': (0, 255, 2
55), 'broccoli': (0, 255, 255), 'carrot': (0, 255, 255), 'hot dog': (0, 255, 255), 'pizza': (0, 255, 255), 'donut': (0, 255, 255), 'cake': (0,
255, 255), 'chair': (0, 255, 255), 'sofa': (0, 255, 255), 'pottedplant': (0, 255, 255), 'bed': (0, 255, 255), 'diningtable': (0, 255, 255), 'to
ilet': (0, 255, 255), 'tvmonitor': (0, 255, 255), 'laptop': (0, 255, 255), 'mouse': (0, 255, 255), 'remote': (0, 255, 255), 'keyboard': (0, 25
5, 255), 'cell phone': (0, 255, 255), 'microwave': (0, 255, 255), 'oven': (0, 255, 255), 'toaster': (0, 255, 255), 'sink': (0, 255, 255), 'refr
igerator': (0, 255, 255), 'book': (0, 255, 255), 'clock': (0, 255, 255), 'vase': (0, 255, 255), 'scissors': (0, 255, 255), 'teddy bear': (0, 25
5, 255), 'hair drier': (0, 255, 255), 'toothbrush': (0, 255, 255)}

Fig. 6; Trained Dataset

{'person': [], 'bicycle': [], 'car': [], 'motorbike': [], 'aeroplane': [], 'bus': [], 'train': [], 'truck': [], 'boat': [], 'traffic light':
[], 'fire hydrant': [], 'stop sign': [], 'parking meter': [], 'bench': [], 'bird': [], 'cat': [], 'dog': [], 'horse': [], 'sheep': [], 'cow':
[], 'elephant': [], 'bear': [], 'zebra': [], 'giraffe': [], 'backpack': [], 'umbrella': [], 'handbag': [], 'tie': [], 'suitcase': [], 'frisbe
e': [], 'skis': [], 'snowboard': [], 'sports ball': [], 'kite': [], 'baseball bat': [], 'baseball glove': [], 'skateboard': [], 'surfboard':
[], 'tennis racket': [], 'bottle': [], 'wine glass': [], 'cup': [], 'fork': [], 'knife': [], 'spoon': [], 'bowl': [], 'banana': [], 'apple':
[], 'sandwich': [], 'orange': [], 'broccoli': [], 'carrot': [], 'hot dog': [], 'pizza': [], 'donut': [], 'cake': [], 'chair': [], 'sofa': [],
'pottedplant': [], 'bed': [], 'diningtable': [], 'toilet': [], 'tvmonitor': [], 'laptop': [], 'mouse': [], 'remote': [], 'keyboard': [], 'cell
phone': [(309, 79, 418, 175)], 'microwave': [], 'oven': [], 'toaster': [], 'sink': [], 'refrigerator': [], 'book': [], 'clock': [], 'vase': [],
'scissors': [], 'teddy bear': [], 'hair drier': [], 'toothbrush': []}

Fig. 7; Detected Objects are Numbered in the Format of center x and y, Width and Height

Further the obtained labels are converted to voice using Python library and hence the final output which is in the form of a voice will be heard through headphones by the visually impaired person. The objects within bounding box and labelled object using machine learning algorithm are shown in Fig. 8.



(a) (b)

(c) (d)

Fig. 8; Detection of (a) Mouse (b) Laptop (c) Clock (d) Cell Phone

CONCLUSION

In this project, we have used the YOLO algorithm for object detection. This approach is generalised, and it outperforms other strategies when applied to diverse domains from live video. The algorithm is simple and trained on an entire image. We have used a trained database of about 80 datasets and look forward to training more datasets in the future. In order to forecast limits full frames of video is used in YOLO. Fewer false positive shall appear in the background. This algorithm is substantially more efficient and faster to employ in real time when compared to other classification algorithms. We want to continue our work in the future by creating our own benchmark dataset and designing a product which would be handy and cost efficient.

REFERENCES

Khairnar, Devashish Pradeep, et al. "Partha: A visually impaired assistance system." 2020 3rd International Conference on Communication System, Computing and IT Applications (CSCITA). IEEE, 2020.

Kaluwahandi, Sasadara, and Yoshiaki Tadokoro. "Portable traveling support system using image processing for the visually impaired." Proceedings 2001 International Conference on Image Processing (Cat. No. 01CH37205). Vol. 1. IEEE, 2001.

Jakhete, Sumitra A., et al. "Object Recognition App for Visually Impaired." 2019 IEEE Pune Section International Conference (PuneCon). IEEE, 2019.

Chung, Il Yong, Sanghag Kim, and Kang Hyeon Rhee. "The smart cane utilizing a smart phone for the visually impaired person." 2014 IEEE 3rd Global Conference on Consumer Electronics (GCCE). IEEE, 2014.

Jain, Sambhav, et al. "Design and implementation of the smart glove to aid the visually impaired." 2019 International Conference on Communication and Signal Processing (ICCSP). IEEE, 2019.

Adarsh, Pranav, Pratibha Rathi, and Manoj Kumar. "YOLO v3-Tiny: Object Detection and Recognition using one stage improved model." 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS). IEEE, 2020.

Ullah, Md Bahar. "CPU based YOLO: a real time object detection algorithm." 2020 IEEE Region 10 Symposium (TENSYMP). IEEE, 2020.

Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." Proceedings of the IEEE conference on computer vision and pattern recognition, 2016.

Rizan, Tharik, et al. "Guided Vision: A High Efficient and Low Latent Mobile App For Visually Impaired." 2021 3rd International Conference on Advancements in Computing (ICAC). IEEE, 2021.

Padmapriya, V., et al. "A Study on text Recognition and Obstacle Detection Techniques." 2020 International Conference on System, Computation, Automation and Networking (ICSCAN). IEEE, 2020.

Caesar, Holger, Jasper Uijlings, and Vittorio Ferrari. "Coco-stuff: Thing and stuff classes in context." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

Mahendru, Mansi, and Sanjay Kumar Dubey. "Real Time Object Detection with Audio Feedback using Yolo vs. Yolo_v3." 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence). IEEE, 2021.

Nasreen, Jawaid, et al. "Object Detection and Narrator for Visually Impaired People." 2019 IEEE 6th International Conference on Engineering Technologies and Applied Sciences (ICETAS). IEEE, 2019.

Kumar, Ashwani, SS Sai Satyanarayana Reddy, and Vivek Kulkarni. "An object detection technique for blind people in real-time using deep neural network." 2019 Fifth International Conference on Image Information Processing (ICIIP). IEEE, 2019.

Abraham, Leo, et al. "VISION-wearable speech based feedback system for the visually impaired using computer vision." 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI) (48184). IEEE, 2020.

Ibrahim, S., & Rababah, A. (2022). Decomposition of Fourth-Order Euler-Type Linear Time-Varying Differential System into Cascaded Two Second-Order Euler Commutative Pairs. Complexity, 2022.