Design and Development of a Smart Eye Wearable for the Visually Impaired

Tamoghna Sarkar, Anith Patel, and Sridhar P. Arjunan

Department of Electronics and Instrumentation Engineering,
SRM Institute of Science and Technology, Chennai 603203, TN, India
ts3044@srmist.edu.in

Abstract. Accessibility and mobility are two of the major domains with which the Visually Impaired are still struggling. In the 21st century, where everything is termed "SMART", we are yet to reach the acme where we can solve the above problems for the Visually Impaired. People today live in smart homes where everything in the house is connected to a common network. Voice assistants are becoming common in every house and besides, Wearable Technology has taken off in diverse directions which were once considered impossible. This has led to an overall paradigm shift in how humans interact with technology. In this paper, we propose a prototype of an assistive attachable that would help the Visually Impaired, for navigation and orientation. The device has the state-of-the-art implementation of Artificial Intelligence on the edge, Computer Vision and Neural Networks. It performs Real-time Image Cognition frame by frame using the camera on the device, undergoes pre-processing of the images on the edge device and performs classification on our trained region convolutional neural network (R-**CNN**). After Image Recognition is successfully performed, the key features of the surroundings are read out into the ears of the Visually Impaired person through audio feedback. It is expected to provide guided navigation, object information about places, products, and services that are present in the vicinity of the user. The results from the data collected and accuracy has been significantly improved with a recognition accuracy of 96%. The proposed smart wearable device has tested in real-time to prove its usefulness for the Visually Impaired.

Keywords: Visually Impaired · Object Detection · Edge Computing · Mask R-CNN · Artificial Intelligence · Localisation · 3D modelling · Audio feedback · Inter-IC-Sound Interface

1 Introduction

Vision is an important aspect of every human throughout a lifetime. According to the World Health Organization, an estimated 2 billion people globally are visually impaired or are totally blind. Out of the 2 billion people, the classifications are: -

On a global scale, the leading causes for vision impairment and blindness include stroke or transient ischemic attack, glaucoma, detachment of the retina, and so on. By the end of 2020, these numbers are predicted to increase exponentially and get doubled [1].

Therefore, visual impairment can be classified into two parts. Partial Vision Impairment: - wherein a person is blind up to a certain percentage and able to perceive

C. Badica et al. (Eds.): ICICCT 2020, CCIS 1170, pp. 208–221, 2020.
https://doi.org/10.1007/978-981-15-9671-1_18

the surroundings in a complete hazy manner and Complete Vision Impairment or a state of complete Blindness.

People distressed with either of them are left in a state of torment and it's an arduous job for them in everyday life as they have to be dependent on others. Accessibility and mobility have always been the utmost conundrums for the visually impaired. Devices existing in the market today are maybe smart assistive devices but evidently not enough to help reduce the pain points of a commensurate, considerable amount of visually impaired people.

Three categories of devices are being developed in order to help Vision Substitution. Vision Substitution: An assistive electronic device with sensors and cutting-edge technology that functions by providing route guides and tries to embellish the mobility of visually impaired people. The three categories for classic electronic devices for the visually impaired are Electronic Travel Aid (ETA), Electronic Orientation Aid (EOA), Position Locator Devices (PLD). These are devices that use SONAR, LASER, sensors for accumulating information about the surroundings, help in pothole detection, gives their location coordinates. In easy words, they are assistive devices [2].

Haptic feedback is used in many of these devices as an alert system. However, these devices have several disadvantages out of which the major ones are stated. Either the devices are too costly which most people can never afford to buy or these devices have a robust size and are too bulky to be carried around. Hence, we cannot totally say that we have been successful in bringing a notable change in the life of the Visually Impaired people. Auditory and Tactile sensors can be deployed for totally blind people. The tactile sensors do not form any obstructions for the auditory sensors, which is the most important unit of perception in every human. Despite that, this approach is avoided due to varied drawbacks and instead of robust sensors we may approach using the synthetic voice or sound as a feedback signal into the ears. This paper is a result of such an approach and an initiative to implement state-of-the art research for the development of a non-intrusive smart attachable in the glasses for visually impaired people [3].

The rest of this paper is organized as follows. In Sect. 2, the Literature Survey is described. Section 3 introduces Materials and methods which includes 3D Modelling of the envisioned attachable, Hardware Platform, the Embedded Detection System on the edge device and the use of Mask R-CNN, followed by experimental results in Sect. 4 and finally conclusions are drawn.

2 Literature Review

Standing in the 21st century, the smart world is one of the most important motivations for most developing countries. We are imminent about amalgamating artificial and penetrative intelligence, Internet-of-Things into ubiquitous things including human behaviour, physical objects around us, social people. Exponentially, we are reaching that state where everything around is ubiquitously intelligent. Without ubiquitous computing, it is very difficult to maintain service support and ambient intelligence. We talk about smart agriculture, smart waste management systems, smart traffic management systems, smart homes, smart devices or wearables. An attempt to convert every ordinary device into a smart with ubiquitous computing is the paramount issue. The leading industries in the world, whether it is an Information Technology company to a core Industry based power plant, everybody tends towards intelligent devices and automation with a motivation to make human lives more comfortable, convenient and informed [4].

Amidst smart devices for daily usage of the normal people, there has also been intense, significant research and development going on for the specially challenged people. Specially challenged people are a part of our society and it's our utmost responsibility that when we think about smart devices all around us, why not implement state-of-the art technology and deploy devices that would make their living smart, comfortable, independent and allow them to have a more immersive experience of their surroundings. There is no usefulness of cutting-edge technology if there is no giving back to society. Some example of some significantly important projects are: -

"Development of a Smart Cane" - Wahab et al. [2] was the chief person behind this research wherein the cane was used mainly for object detection and providing accurate instructions for navigation. Similar, original work was presented by students from Central Michigan University. The system consists of ultrasonic sensors, a microcontroller, a buzzer, and a water detector. It utilizes ultrasonic sensors, fuzzy controllers to detect obstacles around the person and then transfer it to the individual's ears using haptic feedback in the form of vibrations or audio feedback. The cane is basically a sensor-equipped portable device for navigation purposes of the blind people. This device though being able to detect obstacles is not totally non-intrusive as it is robust. It's like the generic canes that the blind people, the only difference being that it has sensor circuitry which makes or smart. The power consumption versus time relationship of the device while the person is on the move is something that has to be looked into [5].

Another prototype, "Fusion of Artificial Vision" demonstrated how Bumblebee stereo cameras were installed on a helmet for video input [6]. The video stream on being captured was processed using the SpikNet recognition Algorithm to locate visual features. These visual features were mapped with GPS coordinates for allowing location services [7].

The Global Position System (GPS), modified Geographical Information Systems (GIS) and vision-based positioning have been implemented for fast localization.

This device comes into the category of Head Mounted Devices for blind but again has a major drawback as its robust structure because the entire setting is done on a helmet that is worn by the blind person. Also, the system has not been tested in actual navigation systems, therefore whether it will ameliorate the navigation system is still unknown.

LASER and SONAR systems have been used with micro-controllers to calculate distances between the object and the obstacle. In addition to that "Obstacle Avoidance using auto-adaptive and Thresholding techniques" using the Kinect depth camera has been tried. But Kinect displayed low accuracy in a short-range and this could directly affect the performance of the system. From the results, it was found out that the auto-adaptive threshold was not able to find distinctions between different objects. Computer Vision methodologies have also been implemented in many research projects for Object Detection, Pothole Detection and so on.

In this paper, we have tried to dive deep into the research aspects, and come up with a Smart Eye Wearable which is expected to solve the drawbacks in most of the mentioned research projects above.

3 Material and Methods

3.1 Three-Dimensional Modelling

Three-dimensional models are 3d-printed using PLA (Polylactic Acid) material for prototyping purposes. We modelled this wearable device for a more effective presentation of the final idea, whether the components we are using fits properly inside the model estimate how much a part or assembly will weigh and obtain its Centre of Gravity. It saves a lot of back and forth wastage in the process of designing final products. Lastly, an idea is always better conveyed when we can present the model to a reviewer physically, instead of trying to make him visualize a vague imaginary picture of the prototype.

The model shown in Fig. 1. is designed in Autodesk Fusion 360. The final project outcome is envisioned to look alike. Due to the already existing availability of assistive devices for the visually impaired that are bulky and are not easy to be carried around while commuting, we decided not to converge towards designing an entirely new device for them, for example, a smart cane, smart helmet with cameras. Instead, design a smart attachable or a wearable that could easily be attached to the spectacles without any extra-hassle. The sleek device attached to the left temple arm and hinges of the rendered design below is the attachable we designed. It is light and could be placed into a purse or pockets eradicating the purpose of carrying something along (Fig. 2).

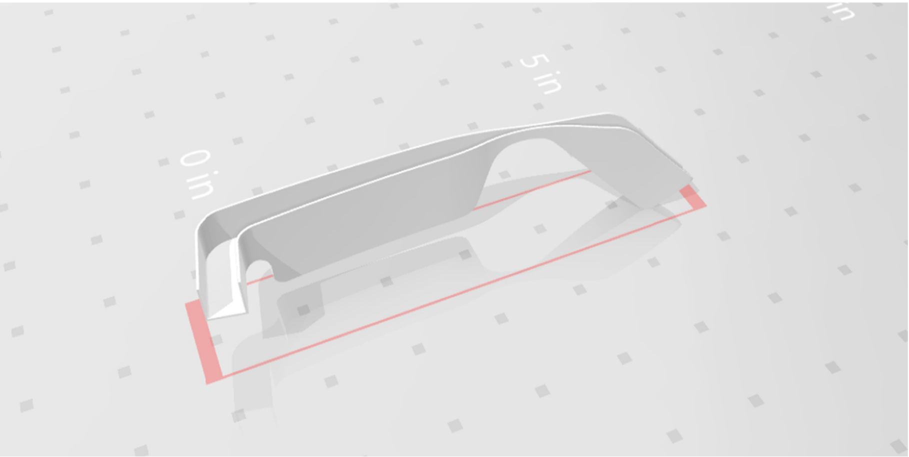


Fig. 1. 3D design of the proposed attachable

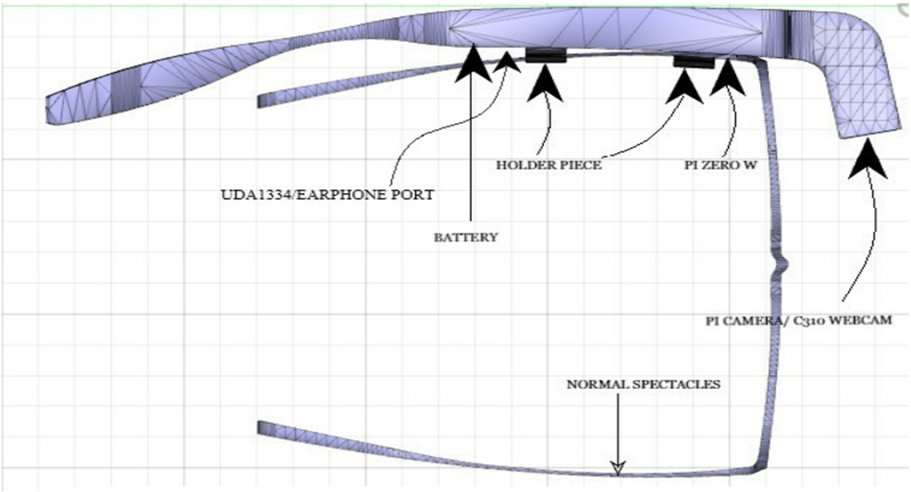


Fig. 2. Labelled representation of components in their envisioned sections

The attachable would consist of the edge device which is Raspberry Pi (R-pi) Zero W in our case. This R-pi would be present vertically along the walls of the device with which a camera would be interfaced. The camera is denoted by small concentric circles in front of the lens of the spectacles. The entire work in this paper is based on Artificial Intelligence on the edge.

3.2 Prototype Hardware Platform

Before we funnel down into algorithms and results for edge computation and object detection, classification techniques, let us discuss the hardware platform which forms the kernel of the remaining portions of this paper. In the rest of the paper, we will delineate the experiments and results obtained to support the theory of computation on edge Artificial Intelligence based on this embedded platform and feasibility of this research project into a full-scaled product that could potentially bring a massive change in the society of the visually impaired.

The proposed attachable for the blind would consist of a Raspberry Pi Zero Wireless, Adafruit I2S Stereo Decoder- UDA1334A, an AC Power supply, a camera, and a PC which behaves like a controller. The camera chosen for the experiments was Pi Camera and a Logitech C310. Both of them can demonstrate almost equal results, except that the C310 has a resolution of 5 megapixels whereas the Raspberry Pi Camera Version 2 has a resolution of 8 megapixels along with Sony IMX219 in-built. The Pi cam has a faster frame per second rate which is 90 fps. Frame acquisition takes place as the camera is connected and powered with the Pi Zero board. The camera captures frame by frame, everything in its vicinity and pre-processes those images on the development board itself. Computer Vision was implemented in the Pi for this purpose. Earlier projects with Computer Vision and object detection have been executed using the Pi 3 and all the higher models, but we chose the Pi Zero W version since we actually designed a 3D model of the attachable and wanted to keep the device lightweight and test the hardware and software in real-time situations. The Pi Zero W has features like 802.11n Wireless LAN and Bluetooth, along with its small size which makes it the best fit for a wearable or attachable development platform.

The board has a 1 gigahertz, single-core Central Processing Unit and therefore it is definite that training the model at the edge or the node is not possible. A GPU is being used for training the model about which we will be discussing in a later section. After Image Captioning is exhibited, the main motive is to provide audio feedback through earphones into the ears of the visually impaired person. To perform this task, the board should be having an I2S interface used for transferring digital audio data between chips. It is also known as the "Inter IC Sound Interface". The RAW/PCM data can be stored on the memory of the micro-controller if the sound file is small and then processed out to the I2S port. The I2S requires accurate clock pulses for working with data back and forth. Though Pi Zero W does not have a direct audio output port like the higher versions, it does support I2S audio output standards on the board. Reconsidering the use case and its size aspects we opted for Adafruit I2S Stereo Decoder-UDA1334A. It is a DAC that processes the data immediately and produces a clear, Analog, stereo line-level output. The pins required for the audio interface of the UDA1334 with the Pi are power pins, ground, DIN, BCLK, LRCLK. The circuit diagram demonstrates the connections clearly between the Pi, UDA1334, the rest of the circuit consists of a camera interfaced with the Pi Zero W and power supply that is connected to the Pi Zero. Finally, the audio output is derived from the audio port of the UDA1334 breakout DAC (Fig. 3 and Fig. 4).

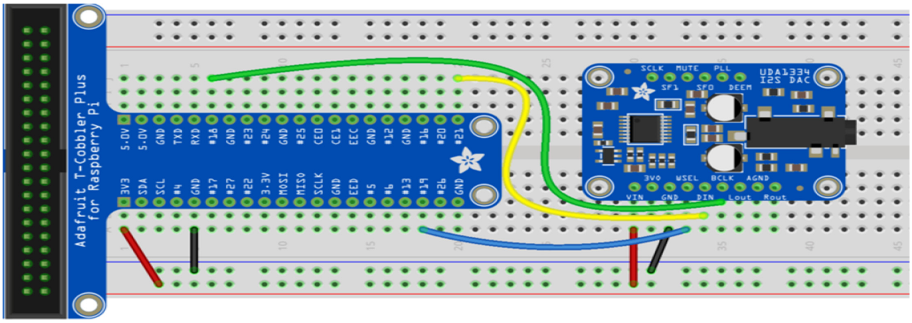


Fig. 3. Circuit connections between Pi Zero W and UDA1334A [https://learn.adafruit.com/adafruit-i2s-stereo-decoder-uda1334a/raspberry-pi-wiring]

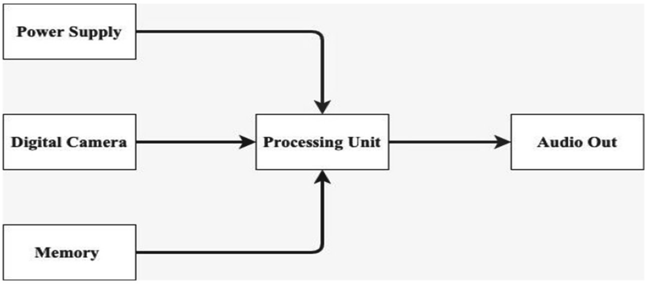


Fig. 4. Block diagram of the system

3.3 Embedded Computation

We are harnessing the power of the edge to perform the computation on a real-time basis. We run the above-mentioned model solely on the embedded computer, in this case, we are using Raspberry Pi, implementing the following flow for the implementation of the first we load the trained model i.e. is trained on the GPU(Graphics Processing Unit), in our case we have used Nvidia TITAN X GPU, usage of the system for training varies from one user to another user. After training, we transfer the weights on the edge device the minimum requirements for this model are 1 GHz of processing and above. Using the system having processing power lower than mentioned can result in the haphazard results. After loading the weights, extraction of the features from the image and conversion of those features into lists and then into string texts for feature to audio takes place, thus the entire process of converting an image to text and text to audio is enchanted here. Since, training of the model is being done on different systems and implementation on another device, the accuracy of predicting the images on the edge is high (Fig. 5).

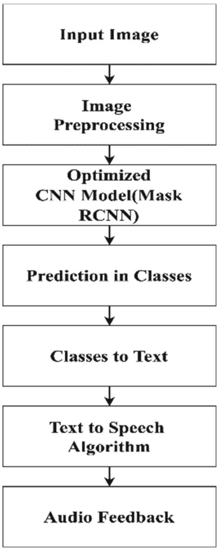


Fig. 5. Flow model of software computation

Object Detection Method. Computer Vision has gained traction in recent years, showing a variety of applications in self-driving cars, security analysis and many more. One of the major applications of this technology is object detection. Object detection aids in pose estimation, surveillance, vehicle detection and so on. Year after year the accuracy of the object detection methods has been increasing exponentially. Various methods and algorithms are tried and tested, like convolutional neural network (**CNN**), region convolutional neural network (R-**CNN),** You Only Look Once (**YOLO**), and so on. For this study, we are implementing the Mask-RCNN framework since it is giving a better overall accuracy in object detection and prediction. Mask RCNN is the extension to the Faster R-CNN method, it adds up the mask as a label to the image giving more accuracy for the output. For our application, we are deploying this model on edge for the real-time detection of the object.

Mask R-CNN. Mask R-CNN uses the two-stage procedure, where the first stage is the same as RCNN i.e. Region Proportion Network (RPN). The second stage is in parallel to predict the class and box, also gives an output in binary for Region of Interest (ROI). This is complementary to the most recent systems where the classification is based on mask predictions.

Formally the loss during the sampling is defined as:

$$L = L_{cls} + L_{box} + L_{mask}$$

The class loss (Lcls) and the bounding box loss (Lbox) are identical as they are defined in Faster-RCNN as well [8]. The mask has Km2 dimensional output for each RoI, which encodes K binary masks of resolution m × m, one for each of the K classes. Here per-pixel sigmoid has been applied and Lmask defined as the average binary cross-entropy loss. For an RoI associated with ground-truth class k, Lmask is only defined on the k-th mask (other mask outputs do not contribute to the loss). L function is optimised in Mask RCNN over other methods so that we can get better and fast outputs (Fig. 6 and Fig. 7).

The masks with a per-pixel sigmoid and a binary loss do not compete with the class and model loss. This is proved in this literature (Fig. 8).

The backbone architecture uses network-depth-features. We evaluate ResNet (Link the citation) and ResNeXt [link the citation] networks having a depth of either 50 or 101 layers. The original implementation of Faster R-CNN with ResNets extracted features from the final convolutional layer of the 4-th stage, which is called C4. Thus, a backbone with ResNet-101, for example, is denoted by ResNet-101-C4.

Implementation. Hyperparameters are set for the existing Faster R-CNN model [9]. We are building on the ImageNet data by adding our customised data set for increasing accuracy according to the environment around us.

Training. RoI is considered to be positive if the ground truth factor is minimum 0.5, thus the mask loss L_{mask} is defined only for positive RoIs. The image pixels are set as 600 × 800 for the training. This method is an image-centric method where the data is predicted based on the image data as the core. The mini-batch has 2 images/GPU and all the images have N sampled RoIs, a ratio of 3:1 of negatives to positives is

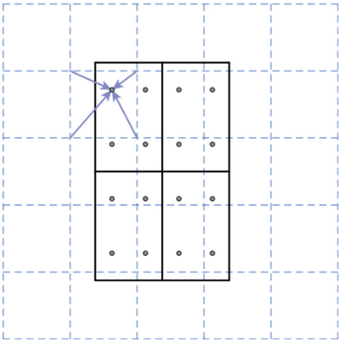
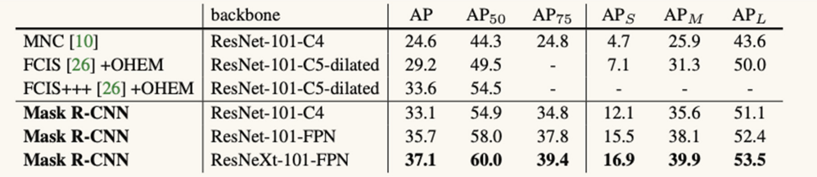


Fig. 6. RoIAlign: the dashed grid represents a feature map, the solid lines an RoI, and the dots the 4 sampling points in each bin. RoIAlign computes the value of each sampling point by bilinear interpolation from the nearby grid points on the feature map [https://arxiv.org/abs/1703.06870]



	backbone	AP	AP$_{50}$	AP$_{75}$	AP$_S$	AP$_M$	AP$_L$
MNC [10]	ResNet-101-C4	24.6	44.3	24.8	4.7	25.9	43.6
FCIS [26] +OHEM	ResNet-101-C5-dilated	29.2	49.5	-	7.1	31.3	50.0
FCIS+++ [26] +OHEM	ResNet-101-C5-dilated	33.6	54.5	-	-	-	-
Mask R-CNN	ResNet-101-C4	33.1	54.9	34.8	12.1	35.6	51.1
Mask R-CNN	ResNet-101-FPN	35.7	58.0	37.8	15.5	38.1	52.4
Mask R-CNN	ResNeXt-101-FPN	**37.1**	**60.0**	**39.4**	**16.9**	**39.9**	**53.5**

Fig. 7. Mask R-CNN outperforms the more complex FCIS +++, which includes multi-scale train/test, horizontal flip test, and OHEM. All entries are single-model results. This is the data from the COCO dataset where Mask-RCNN outperformed every other model

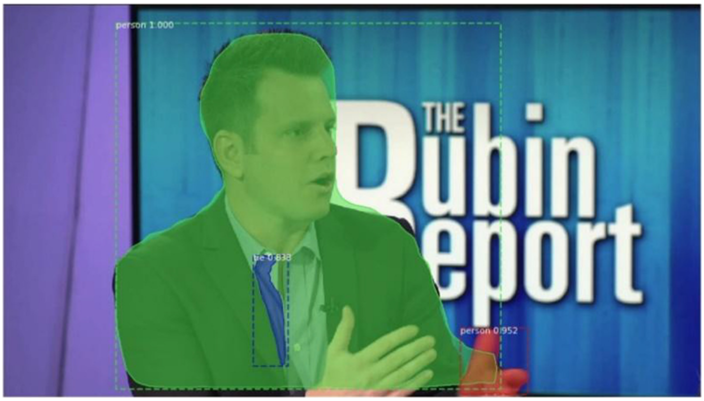


Fig. 8. Model running in action in real-time and returning the object with 1.00 probability

mentioned. The C4 backbone value of N is defined as 64 [10] and for FPN it is 512 [9]. Training takes place on a minibatch size of 16 for 160 k iterations, with a learning rate of 0.02, which is decreased by 10 at the 100 k iteration. Weight decay of 0.0001 and a momentum of 0.9 is used. With ResNeXt [11], 1 image per GPU computation is trained and the same number of iterations, with a starting learning rate of 0.01.

Inference. During testing the backbone number for C4 was 250 and for the FPN it was 1100, thus the box prediction branch is running in parallel that is followed by non-maximum suppression. These Mask branches can predict N masks per RoI which is enhancing the label data. The binarized threshold for this model is 0.5.

Application. We will be utilizing this technology to implement in the proposed model where we will be converting the image data into audio output. Further, in this literature, we discuss the overall implementation of the model in the proposed system.

Text to Speech. In our case, we are using GOOGLE Text-to-Speech. Text-to-speech is one of the most important portions of this research project. This is because it is the final stage of the output phase through which the visually impaired person gets to understand the surrounding.

Furthermore, when the live video is being captured and the frames are being processed, the key features of the frames are being extracted using a classification algorithm that is implemented on our deep neural network. The Mask RCNN is also used in numerous cases to solve segmentation problems in Machine Learning and Computer Vision. After the segmentation and classification of key objects in the frames are done, the classes return values as outputs after execution. Those values are converted into string representations.

These strings are demonstrated in a format, "'There is' + person + 'in the frame'". The rearmost step is the conversion of the string data types into speech files. Speech files can be stored as .mp3 or as .wav format and these sound files are then stored. Hence, the conversion from text to speech is completed after the object detection process. The entire process is automated by our python script. The stored audio file is then transmitted as audio feedback in the ears of the visually impaired person through earphones using I^2S (Inter-IC Sound) transmission.

4 Results and Discussions

This project has the implementation of multiple concepts channelling the flow as Image Recording, Object detection from the image, converting objects into the text which is called Image captioning, converting text to audio using Google Text-to-Speech. To make the above flow functional, we used Raspberry Pi Zero W to run the model and software aspects on the board. For image recognition, we get the output in a span of 1–10 s depending on the number of objects in the image. To achieve this performance,

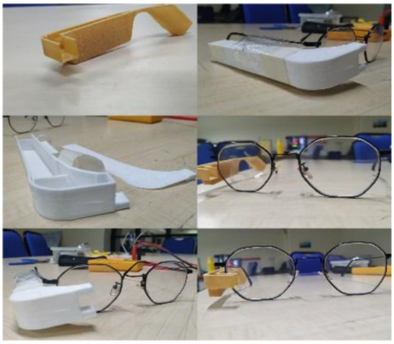
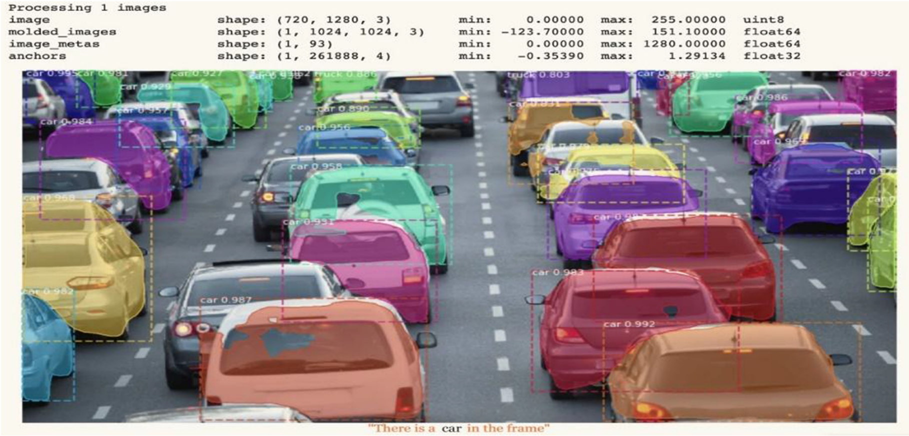


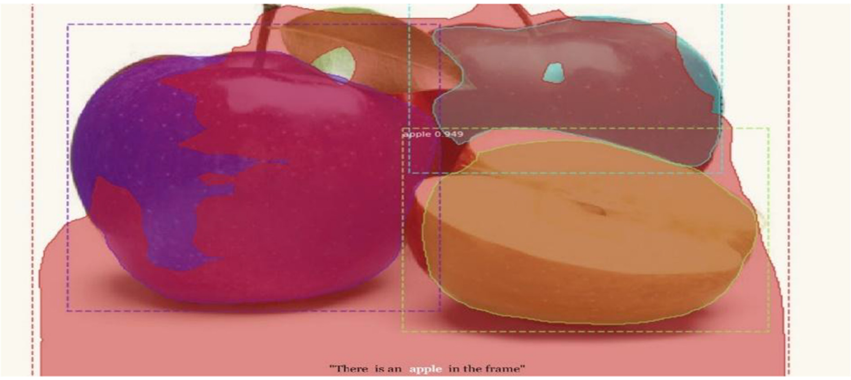
Fig. 9. 3D Printed model along with spectacles



```
Processing 1 images
image                  shape: (720, 1280, 3)        min:    0.00000  max:  255.00000  uint8
molded_images          shape: (1, 1024, 1024, 3)    min: -123.70000  max:  151.10000  float64
image_metas            shape: (1, 93)               min:    0.00000  max: 1280.00000  float64
anchors                shape: (1, 261888, 4)        min:   -0.35390  max:    1.29134  float32
```

"There is a car in the frame"

Fig. 10. Output 1 demonstrating image captioning and classification performed in Mask R-CNN



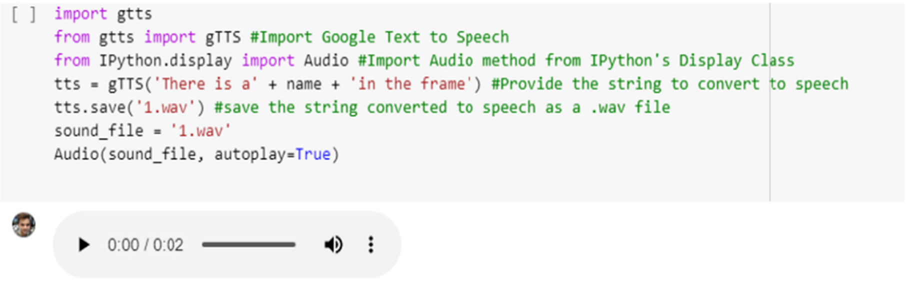
"There is an apple in the frame"

Fig. 11. Output 2 demonstrating image captioning and classification performed in Mask R-CNN

we threaded the neural network to a 1-D array. After the model detects the object from the shutter of images, it converts objects in the image to audio which is achieved in 0.5 s. Thus, making overall system performance ranging from 1.5–10.5 s for performing flow. Along with Raspberry Pi, we use cameras for input and earphones for output (Fig. 9, Fig. 10 and Fig. 11).

4.1 Localization

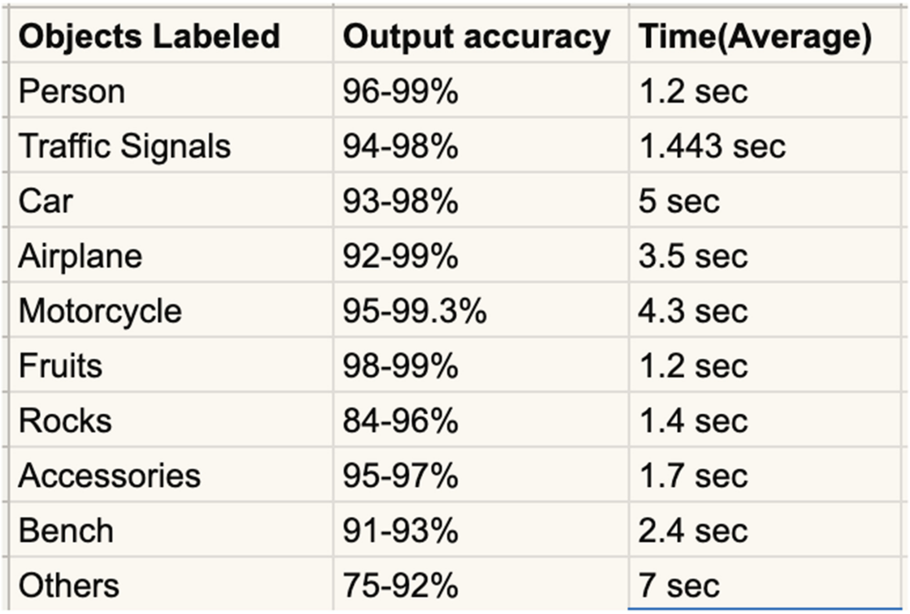
The above figures prove the feasibility of the Mask R-CNN model. Besides, Image Captioning is also being performed successfully where the details present in the image being fed are being converted into text through suitable localization and segmentation. Figure 12 displays the .Wav sound file being produced after objects in the images are detected, classification is performed and details of the image are converted to strings. Next, the string file is converted into an audio file that may be either be in .Mp3 or . Wav format. This audio file is then sent as an output audio feedback through Text-to-speech services. The output has to be collected from the audio port of the UDA1334A board through an earphone. Therefore, despite the presence of some existing similar research in this area, we claim that the Mask R-CNN model we are using here sees an overall increase in accuracy up to 96%. Reduced latency, computation of the whole process is observed, and power consumed to perform the experiments were very low due to the low power requirement of the Pi Zero board.



```
[ ]  import gtts
     from gtts import gTTS #Import Google Text to Speech
     from IPython.display import Audio #Import Audio method from IPython's Display Class
     tts = gTTS('There is a' + name + 'in the frame') #Provide the string to convert to speech
     tts.save('1.wav') #save the string converted to speech as a .wav file
     sound_file = '1.wav'
     Audio(sound_file, autoplay=True)
```

Fig. 12. Text-to-speech execution and generation of .Wav file

Figure 13 demonstrates the quantitative data for the experiment being performed on a real time basis, the model is trained on 5000 images. And we got the output as follows



Objects Labeled	Output accuracy	Time(Average)
Person	96-99%	1.2 sec
Traffic Signals	94-98%	1.443 sec
Car	93-98%	5 sec
Airplane	92-99%	3.5 sec
Motorcycle	95-99.3%	4.3 sec
Fruits	98-99%	1.2 sec
Rocks	84-96%	1.4 sec
Accessories	95-97%	1.7 sec
Bench	91-93%	2.4 sec
Others	75-92%	7 sec

Fig. 13. Quantitative analysis of testing data

From the above results we get the glance of the model working on the system. Where the accuracy produced is reasonable and if more data points for the given samples are used, it could result in increasing the accuracy. In the others section there were various other labels used like ‘suitcase’, ‘frisbee’, ‘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’, ‘couch’, ‘potted plant’, ‘bed’, ‘dining table’, ‘toilet’, ‘tv’, ‘laptop’, ‘mouse’, ‘remote’, ‘keyboard’, ‘cell phone’, ‘microwave’, ‘oven’, ‘toaster’, ‘sink’, ‘refrigerator’, ‘book’, ‘clock’, ‘vase’, ‘scissors’, ‘teddy bear’, ‘hair drier’, ‘toothbrush’. etc. more details of the labels can be found in the code.

5 Conclusion

As mentioned earlier in the result, we are able to run the model on a real-time basis on the edge, where we are getting output in the range of 1.2 to 10.5 s, depending on the number of objects in the frame. The model that is implemented, is highly accurate and results to an output accuracy on testing images at around 96%. Apart from Mask-RCNN other techniques were implemented as well where the accuracy and computation time differed. We selected Mask-RCNN over others for the ease of accuracy and time. This device is in the testing phase where the proof of concept works with good efficiency. In this project, we prototyped a device for visually impaired that could have the potential to help them in mobility and orientation related problems. We implemented embedded Artificial Intelligence and Edge Computing to make this device. This device as of now performs conversion of image to text and gives audio output, for further development we plan to integrate OCR (Optical Character Recognition), making an RTOS that is customised for performing activities especially for the blind, add navigation using haptics for ease in mobility. Thus, the overall technology stack of Embedded AI, CNN, and real time systems can help cater visually impaired to enhance their mobility experience.

All the code to this technology is open-sourced and can be found at (www.github.com/anithp/smartglass).

References

1. Tian, L., Tian, Y., Yi, C.: Detecting good quality frames in videos captured by a wearable camera for blind navigation. In: 2013 IEEE International Conference on Bioinformatics and Biomedicine, Shanghai, pp. 334–337 (2013)
2. Elmannai, W., Elleithy, K.: Sensor-based assistive devices for visually-impaired people: current status, challenges, and future directions. https://www.ncbi.nlm.nih.gov/pubmed/?term=Elleithy%20K%5BAuthor%5D&cauthor=true&cauthor_uid=28287451
3. Bai, J., Lian, S., Liu, Z., Wang, K., Liu, D.: Smart guiding glasses for visually impaired people in indoor environment. https://arxiv.org/ftp/arxiv/papers/1709/1709.09359.pdf
4. Ma, J., Yang, L.T., Apduhan, B.O., Huang, R., Barolli, L., Takizawa, M.: Towards a smart world and ubiquitous intelligence: a walkthrough from smart things to smart hyperspaces and UbicKids. Int. J. Pervasive Comput. Commun. **1**(1), 53–68 (2005)
5. Wahab A., et al.: Smart cane: assistive cane for visually-impaired people. Int. J. Comput. Sci. **8**, 4 (2011)
6. Brilhault, A., Kammoun, S., Gutierrez, O., Truillet, P., Jouffrais, C.: Fusion of artificial vision and GPS to improve blind pedestrian positioning. In: Proceedings of the 4th IFIP International Conference on New Technologies, Mobility and Security (NTMS); Paris, France, 7–10 February 2011, pp. 1–5 (2011)
7. Loomis, J.M., Golledge, R.G., Klatzky, R.L., Speigle, J.M., Tietz, J.: Personal guidance system for the visually impaired. In: Proceedings of the First Annual ACM Conference on Assistive Technologies; Marina Del Rey, CA, USA. 31 October–1 November 1994 (1994)
8. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8691, pp. 346–361. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10578-9_23
9. Lin, T.-Y., Dollar, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: CVPR 2017 (2017)
10. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: NIPS (2015)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)