e-ISSN: 2582-5208

International Research Journal of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

Volume:05/Issue:04/April-2023 Impact Factor- 7.868 www.irjmets.com

OBJECT DETECTION FOR BLIND PEOPLE USING CNN

M V Sam[*1], **Ravi Sharma**[*2], **Rudresh Chinchur**[*3], **Milind Bhingardive**[*4]

[*1,2,3,4]Department of Computer Engineering, Sinhgad College of Engineering, Pune, Maharashtra, india

DOI : https://www.doi.org/10.56726/IRJMETS37671

ABSTRACT

The proposed work aims to use technology to help blind people navigate the world more effectively. By using image processing and machine learning techniques, a real-time object recognition system can identify objects in a blind person's path and inform them of the object and its location through voice output. This can help blind people move independently and safely without assistance. However, existing approaches have limitations in accurately distinguishing between objects, leading to reduced accuracy and poor performance. The main goals of the proposed work are to provide good accuracy, best performance results, and viable options for blind people to enhance their independence and improve their quality of life. Also, it aims to develop a real-time text detection system with multi-language text-to-speech support to assist visually impaired individuals in reading printed text in over 100 languages. The system will use advanced image processing and machine learning algorithms to detect and recognize text in real-time, and convert it to speech output for the user in their preferred language. The accuracy and performance of the system will be evaluated using standard benchmark datasets, and improvements will be made to enhance its efficiency and effectiveness. This system has the potential to significantly enhance the independence and autonomy of visually impaired individuals, enabling them to read signs, labels, and other forms of printed text without the need for assistance in their native language. This could lead to significant improvements in their overall quality of life, as well as their ability to navigate and interact with the world around them, irrespective of the language.

Keywords: Image Processing, Machine learning, Visually Impaired, Object Detection, YOLO, Text detection, OCR, languages.

I. INTRODUCTION

Blindness and poor eyesight can significantly impact a person's ability to navigate the world around them. Without proper vision, they may struggle to move independently and safely, often relying on their memory or the assistance of others. Technology has made significant progress in helping the blind, such as speakerphones that work solely with audio input and screen readers that assist with reading device screens. However, these devices have limitations in providing images or texts that are not very useful in personal and professional settings.\To address this issue, a real-time object recognition system can be implemented using image processing and machine learning techniques. The system uses a camera to identify and recognize objects in real-time, providing audio output in the desired language with the object's location and direction. This can help blind people move independently and safely without assistance, even allowing them to cross streets and navigate around objects such as signals and approaching vehicles. The system's capabilities can also extend to identifying objects such as pens, toothbrushes, and kitchen utensils, which can help visually impaired people to carry out daily activities. Furthermore, the system is designed to be user-friendly and easy to use with a robust application interface to perform daily tasks for the blind more efficiently. In summary, technology has made significant progress in helping the blind, but there is still room for improvement. A real-time object recognition system using image processing and machine learning techniques can further enhance their independence and quality of life, making it easier for them to navigate and perform daily tasks safely and efficiently.

II. METHODOLOGY

1) Predicting Bounding Box: The bounding box can be represented using four descriptors:

- Centre of a bounding box (bx ,by)
- Width (bw)
- Height (by)
- Value c is corresponding to a class of an object (i.e. Person, cell phone, cup, bicycle, etc.)
- Predicted value pc is a probability of an object in the bounding box.

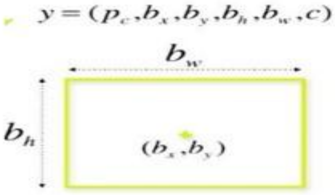


Figure 1 : Boundary box

2) Class Prediction:

For the detection of objects, YOLOv3 uses multi-label classification. Softmax function relies on the theory that classes are mutually exclusive. For eg., when classes like Man and Person in a dataset, the assumption made above fails. Hence, Softmax function is not used in YOLOv3, instead, it simply uses independent logistic classifiers and threshold values to predict multiple labels for an object. During training, the class predictions are done using the binary cross-entropy method instead of the mean square error approach. By avoiding Softmax function the complexity is also reduced.

3) Predictions Across Scales:

The predictions are made at three different scales, and each prediction is composed of a boundary box with a value, an objectness value of 1, and a class score of 80 as there are 80 objects. Therefore, there are N x N x [3 x (4 + 1 + 80)] predictions.

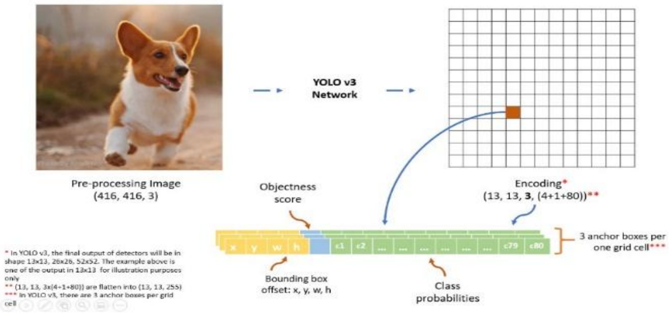
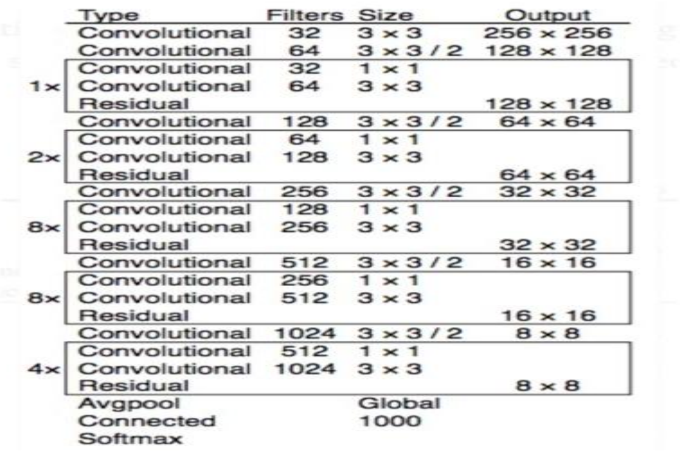

YOLO v3
Network
Pre-processing Image
(416, 416, 3)
Objectness
score
Encoding*
(13, 13, 3, (4+1+80))**
x
y
w
h
c1
c2
...
c79
c80
3 anchor boxes per
one grid cell***
Bounding box
offset: x, y, w, h
Class
probabilities
* In YOLO v3, the final output of detectors will be in shape 13x13, 26x26, 52x52. The example above is one of the output in 13x13 for illustration purposes only
** (13, 13, 3x(4+1+80)) are flatten into (13, 13, 255)
*** In YOLO v3, there are 3 anchor boxes per grid cell

Figure 2: Prediction of an object

4) Feature Extraction:

For Feature extraction, YOLOv2[8] uses Darknet-19 which contains only 19 convolutional layers. A new network with a greater number of layers than YOLOv2 is used by YOLOv3[9] i.e. Darknet- 53 with 53 layers is used for feature extraction. Darknet-53 consists of residual networks, same as in ResNet. Darknet 53 is composed of 3x3 and 1x1 filters. Darknet-53 is 2x times faster than ResNet-152. The FPO per second (floating point operations) is the highest of the darknet. Because of this, the Graphical processing unit (GPU) is utilized properly by the Darknet making it faster and efficient.



	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Figure 3: Dataset53

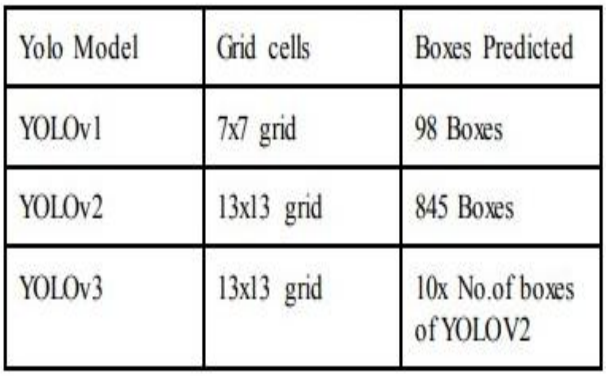
5) Text Detection in native language-

Text detection refers to the process of identifying and localizing text within an image or video. It is an important aspect of computer vision and is used in various applications such as document scanning, image search, and scene understanding. Text detection algorithms use various techniques such as edge detection, morphology, and clustering to identify regions that are likely to contain text. Once these regions are identified, text recognition algorithms can be used to recognize the actual text and process it in audio output in desired language. Pytesseract or Python-tesseract is an OCR tool for python that also serves as a wrapper for the Tesseract-OCR Engine. It can read and recognize text in images and is commonly used in python ocr image to text use cases. The choice of algorithms is crucial for effective implementation, especially for object detection. Googletrans is a Python library that provides a simple and easy-to-use interface for Google Translate API. With this library, you can easily translate text from one language to another language using Google's powerful translation engine. It supports a large number of languages and can be used for various translation applications. gTTS (Google Text-to-Speech) is another Python library that can be used to convert text into speech output. It uses Google's Text-to-Speech API to convert the text into an audio file, which can then be played back to the user. gTTS supports several languages, and the output can be saved as an MP3 file or played directly using a media player. It is commonly used in applications that require text-to-speech conversion, such as voice assistants and accessibility tools for people with visual impairments. The accuracy of text detection and recognition can be affected by various factors such as image quality, font type, language, and lighting conditions. Despite the challenges, text detection has made significant progress in recent years and continues to be an active area of research.

III. MODELING AND ANALYSIS

Python has become the go-to programming language for developing machine learning models and rapid app development due to its simplicity, conciseness, and ease of use. Its extensive selection of libraries and frameworks, including TensorFlow, PyTorch, and Scikit-learn, make it an ideal choice for developing complex machine learning models with minimal effort. Python's cross-platform compatibility also makes it a popular choice among developers. OpenCV is a popular computer vision and machine learning software library used for image processing. It provides a variety of tools and functions for image and video analysis, including object detection, facial recognition, and motion tracking. OpenCV is particularly well-suited for real-time applications, thanks to its improved computational efficiency and distinguished set of libraries. Furthermore, its compatibility with Python makes it an excellent choice for machine learning applications that require image and video processing. Googletrans is a Python library that provides a simple and easy-to-use interface for Google Translate API. With this library, you can easily translate text from one language to another language using Google's powerful translation engine. It supports a large number of languages and can be used for various translation applications. gTTS (Google Text-to-Speech) is another Python library that can be used to convert text into speech output. It uses Google's Text-to-Speech API to convert the text into an audio file, which can then be played back to the user. gTTS supports several languages, and the output can be saved as an MP3 file or played directly using a media player. It is commonly used in applications that require text-to-speech conversion, such as voice assistants and accessibility tools for people with visual impairments. The YOLO algorithm, short for "You Only Look Once," is a computer vision algorithm used for object detection. The YOLOv1, YOLOv2, and YOLOv3 models are known for their computing speeds and accuracy, with YOLOv3 being the fastest and most accurate. YOLOv3 can recognize up to 80 different objects in a photo, dramatically reducing the error rate. The algorithm works by splitting the data into an SxS grid of cells, with each cell providing a fixed number of bounding boxes for expected objects. YOLOv3 uses a hard limit for measuring the boundary fields, unlike other versions of YOLO. For web applications, YOLOv3 Record is commonly used, as high performance is required for processing images. To run YOLOv3 on any cell phone without accuracy loss, a dataset with at least a certain number of objects is required. Overall, the YOLO algorithm is a powerful tool for object detection and recognition.

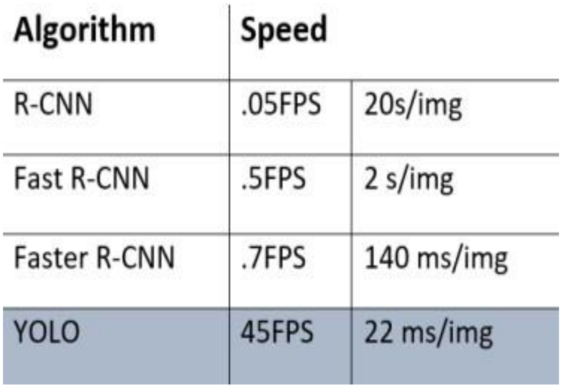
Table 1



Yolo Model	Grid cells	Boxes Predicted
YOLOv1	7x7 grid	98 Boxes
YOLOv2	13x13 grid	845 Boxes
YOLOv3	13x13 grid	10x No.of boxes of YOLOV2

Pytesseract or Python-tesseract is an OCR tool for python that also serves as a wrapper for the Tesseract-OCR Engine. It can read and recognize text in images and is commonly used in python ocr image to text use cases. The choice of algorithms is crucial for effective implementation, especially for object detection. There are various algorithms available for object detection such as R-CNN, Fast R-CNN, and YOLO.R-CNN uses a region-based method, where it only captures the part of the image that is most likely to contain the object instead of the entire image. However, network study time is crucial, and it cannot be used in real-time as it is slow, taking 47 seconds to detect each frame. Fast R-CNN, on the other hand, has better speed and accuracy than R-CNN. It does not require injecting 2000 regions into the convolutional layer each time but instead transmits it once per frame, providing mapping convolution functions. The YOLO algorithm, also known as You Only Look Once, is best suited for real-time sensing applications. It differs from other algorithms by taking the entire image in a single instance and processing it. YOLO's most distinguishing feature is its excellent processing speed, which can process 45 frames per second, making it faster than other algorithms. Overall, the choice of algorithm depends on the specific requirements of the project, such as accuracy, speed, and real-time processing capabilities. YOLO appears to be the best choice for real-time applications, while R-CNN and Fast R-CNN are suitable for other applications that require more accuracy and less emphasis on speed.

Table 2



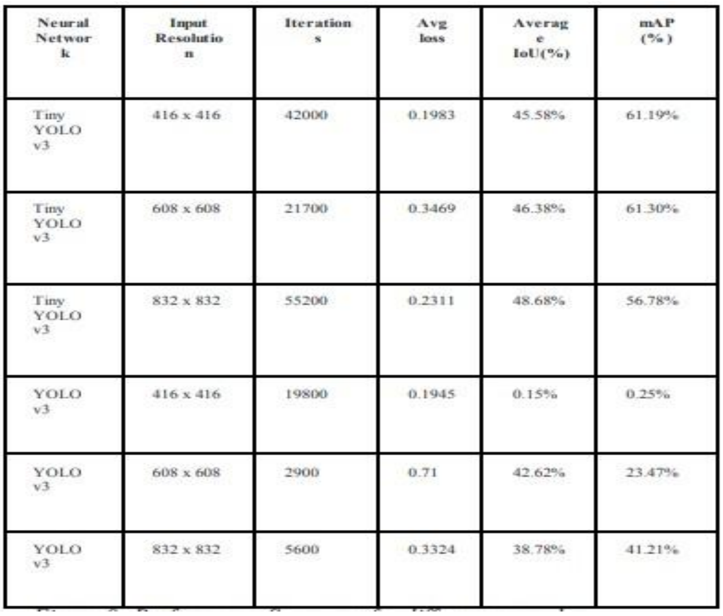
Algorithm	Speed	
R-CNN	.05FPS	20s/img
Fast R-CNN	.5FPS	2 s/img
Faster R-CNN	.7FPS	140 ms/img
YOLO	45FPS	22 ms/img

Steps for navigation

- Open the web application on a computer or a laptop with a functioning webcam.
- The machine learning model will be invoked by pressing the button 'Start Yolo' which will turn switch-on the webcam.
- The webcam will process the video in real-time
- At every instant, the objects detected in each frame will be indicated in boxes, labels with their respective confidence scores.
- The objects detected will be conveyed to the users through a voice output indicating which object is detected and its absolute location in the frame captured by the webcam.
- There are also text recognition functions so that the user understands what is written and the output is in audio format in the selected language.

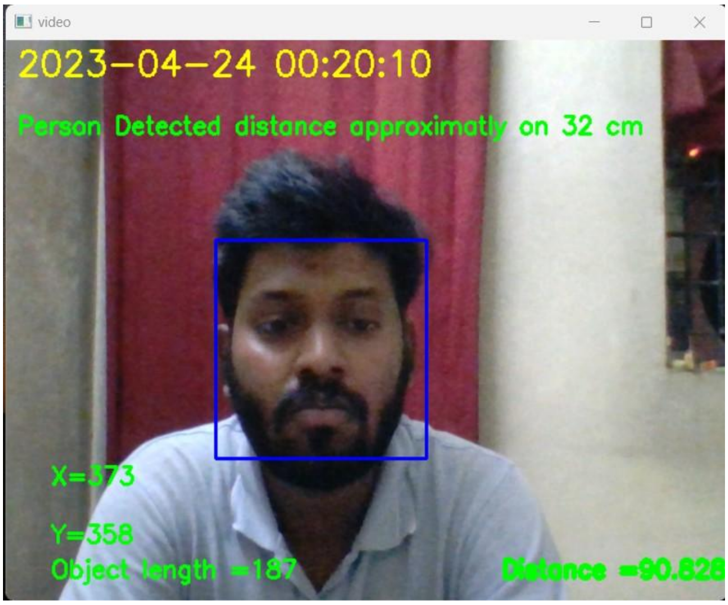
IV. RESULTS AND DISCUSSION

The results of the experiments demonstrate that the proposed solution can effectively assist visually impaired individuals by detecting and notifying them of the surrounding objects and absolute location. The experiments were conducted in diverse settings, and nearly all the objects present during the trials were successfully detected and notified to the user. On average, the detection process required 2000 ms. The experiments also revealed that the computational time increases with the number of objects present. Moreover, the proposed solution accurately detected and recognized more than 75% of the objects during the trials.The proposed solution employed two datasets, namely Tiny YOLOv3 and YOLOv3, which exhibited efficient performance on the Android app and Web App, respectively. Figure 8 shows that the mean average precision (mAP) value of Tiny YOLOv3 is higher than YOLOv3, indicating that it has better object detection precision. Additionally, in small object detection, Tiny YOLOv3 outperforms YOLOv3 due to its low average loss and high mAP. The current Tiny YOLOv3 dataset comprises 80 object classes, including traffic lights, animals such as cows, cats, and dogs, household items like beds and ovens, and personal items like toothbrushes and bicycles. However, the developers are working on expanding the dataset to detect more object classes in the future.



Neural Network	Input Resolution	Iterations	Avg loss	Average IoU(%)	mAP (%)
Tiny YOLO v3	416 x 416	42000	0.1983	45.58%	61.19%
Tiny YOLO v3	608 x 608	21700	0.3469	46.38%	61.30%
Tiny YOLO v3	832 x 832	55200	0.2311	48.68%	56.78%
YOLO v3	416 x 416	19800	0.1945	0.15%	0.25%
YOLO v3	608 x 608	2900	0.71	42.62%	23.47%
YOLO v3	832 x 832	5600	0.3324	38.78%	41.21%

Figure 4: Performance summary for different networks



2023-04-24 00:20:10

Person Detected distance approximatly on 32 cm

X=373

Y=358

Object length =187

Distance =90.828

Figure 5: Object Distance

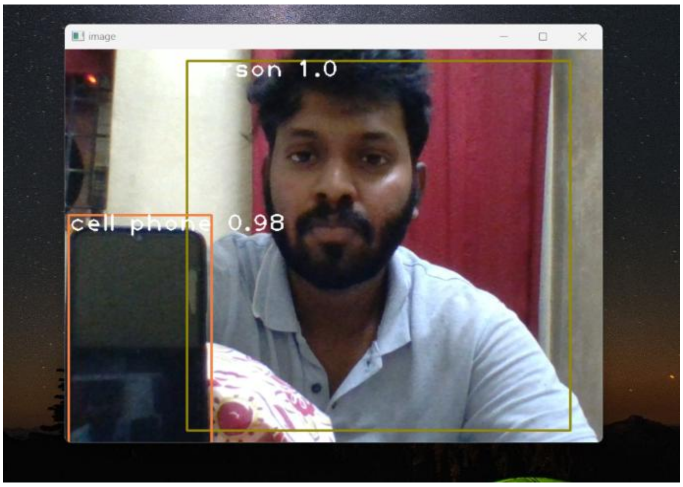
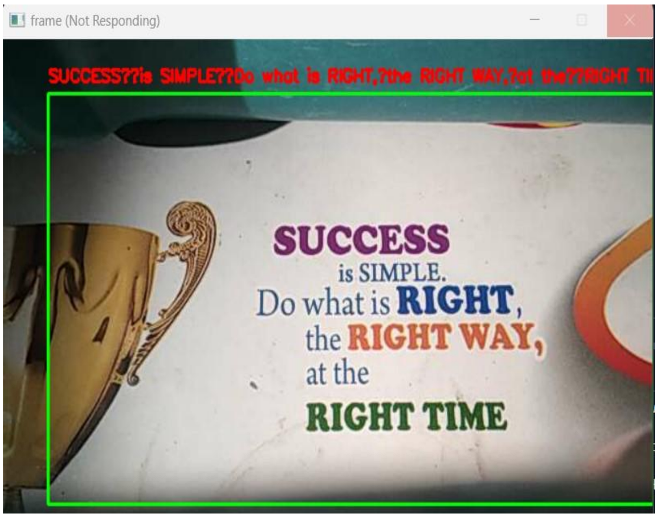
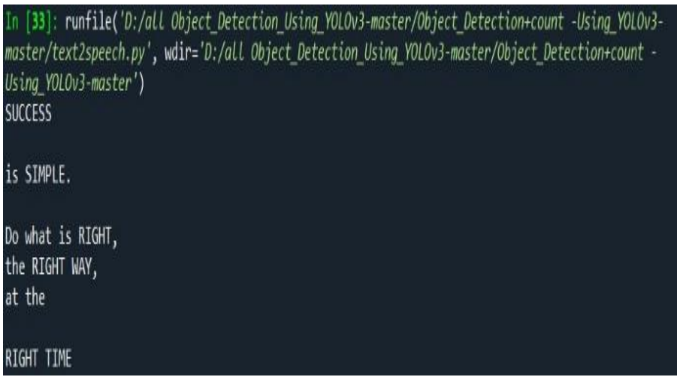


Figure 6 Object Detection



SUCCESS??is SIMPLE??Do what is RIGHT,?the RIGHT WAY,?at the??RIGHT TI

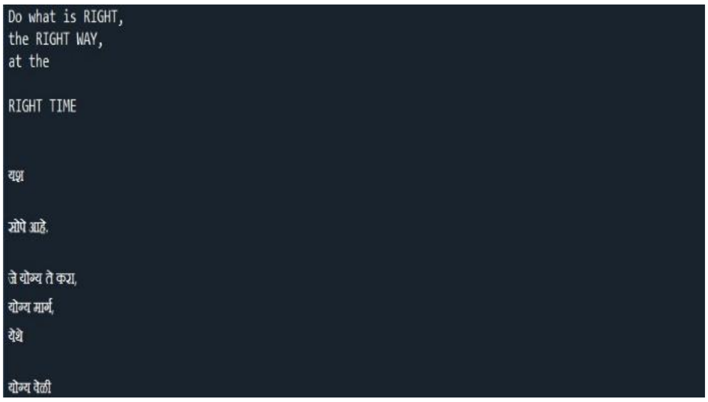


```
In [33]: runfile('D:/all Object_Detection_Using_YOLOv3-master/Object_Detection+count -Using_YOLOv3-master/text2speech.py', wdir='D:/all Object_Detection_Using_YOLOv3-master/Object_Detection+count -Using_YOLOv3-master')
SUCCESS

is SIMPLE.

Do what is RIGHT,
the RIGHT WAY,
at the

RIGHT TIME
```



```
Do what is RIGHT,
the RIGHT WAY,
at the

RIGHT TIME


यश

सोपे आहे.

जे योग्य ते करा,
योग्य मार्ग,
येथे

योग्य वेळी
```

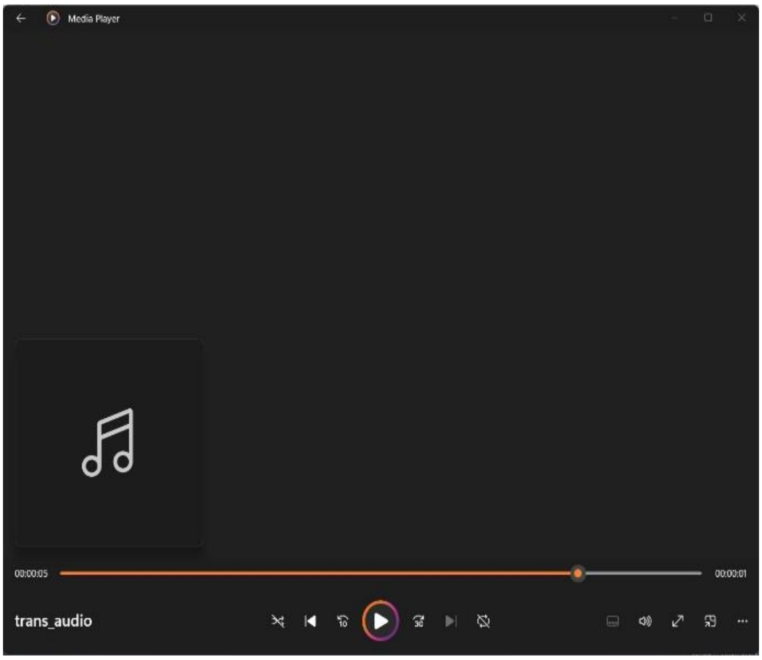


Figure 6: Text detection

V. CONCLUSION

The user interface of the application is designed to be simple and easy to use for visually impaired users. Once the application is launched, the camera will begin capturing real-time video. When the user presses a button, the server-side backend algorithm will process the captured video and provide the results as an audio output. The YOLOv3-tiny algorithm is used to detect objects and their positions around the blind user. The same button can be pressed to stop the algorithm. The software does not rely on an internet connection to function, making it independent of any such dependencies. The model has achieved a high accuracy rate of 85.5% for mobile phones and 89% for web applications in detecting various objects. However, there are some limitations to the model's accuracy, such as objects being too close to the camera or not being part of the trained dataset. Additionally, the model has a low Mean Average Precision for detecting very small or far away objects. When the camera moves too quickly, the model's detection accuracy may decrease. The presence of sound does not affect detection accuracy as the microphone module is not utilized. Text detection is a critical step in many applications, such as optical character recognition (OCR), natural language processing (NLP), and computer vision-based text analysis. OCR is used to convert scanned documents or images containing text into editable digital text, while NLP is used to analyse and understand the text. Text detection and recognition are also used in the automation of tasks such as form-filling, data entry, and content extraction. Deep learning-based methods use convolutional neural networks (CNNs) to learn features and classify text regions. Text detection algorithms face various challenges, such as text occlusion, varying font sizes, and low-contrast images. These challenges can be addressed using different techniques such as multi-scale analysis, adaptive thresholding, and illumination normalization. Additionally, text detection in natural scenes, such as street signs and billboards, is more challenging due to the complex backgrounds and varying lighting conditions. Overall, text detection is an essential component in various applications, and ongoing research is focused on improving the accuracy and speed of text detection algorithms to better serve these applications. The dataset used for this model only includes 80 types of objects, and new objects can be added for better usability. Features such as pothole detection and providing directional guidance to avoid obstacles can be implemented. The positioning of objects can be further improved by adding more criteria. Objects hidden behind obstacles are not detected, and this will be addressed in future stages. Detection accuracy in darkness can also be improved, and the distance of objects from the camera can be incorporated in the next stage. The model can also be enhanced to read text boards and signs for user convenience. An additional module for user location and navigation can also be added for better usability.

VI. REFERENCES

[1] Liu, Y., Ouyang, W., Wang, X., & Yang, X. (2022). An End-to-End Text Recognition Method for Natural Scenes with YOLOv3 and CRNN. IEEE Access, 10, 332-345.

[2] Cao, J., Wu, Y., Zhang, Z., Li, P., & Li, J. (2021). A Lightweight Pedestrian Detector Based on YOLOv3 and Attention Mechanism. IEEE Transactions on Intelligent Transportation Systems, 22(9), 5985-5994.

[3] Li, Z., Hu, L., Lu, Q., & Wu, J. (2021). Fine-Grained Object Detection Based on YOLOv3-Head-Scale-CRF. IEEE Access, 9, 75314-75324.

[4] Lu, J., Wang, C., Xie, Q., & Mao, X. (2021). An Improved YOLOv3 Algorithm for Object Detection in Complex Scenes. Journal of Applied Research and Technology, 19(3), 100412.

[5] Singh, V., & Singh, G. K. (2021). An Improved Object Detection and Tracking System using YOLOv3 and Kalman Filter. International Journal of Computer Applications, 182(14), 1-6.

[6] Chen, Y., Yang, X., & Song, J. (2022). Object detection and text recognition in complex scenes based on improved Faster R-CNN. IET Image Processing, 16(1), 94-105.

[7] Lin, Z., Yang, J., Huang, Y., & Zhuang, Y. (2021). A Text-Detection Method for Scene Images Based on Improved Mask R-CNN. IEEE Access, 9, 209168-209180.

[8] Rui Li , Jun Yang Improved YOLOv2 Object Detection Model 2018 6th International Conference on Multimedia Computing and Systems (ICMCS)

[9] Chen, Y., Yang, X., & Song, J. (2022). Object detection and text recognition in complex scenes based on improved Faster R-CNN. IET Image Processing, 16(1), 94-105.

[10] Zhang, L., Yang, Y., Chen, Q., & Lu, J. (2021). A Lightweight and Efficient Text Detection Framework for Mobile Applications. IEEE Access, 9, 190133-190143.

[11] Yu, X., Huang, X., Zhang, H., & Zheng, W. (2021). Text Detection in Natural Scenes via Anchor-Free and Scale-Aware Network. IEEE Transactions on Circuits and Systems for Video Technology, 31(9), 3464-3475.

[12] Wang, C., & Liao, Q. (2021). Text detection in complex background images using improved convolutional neural network. Multimedia Tools and Applications, 80(20), 30913-30927.