Visual Assist:ML Based Object Detection For the Partially Impaired

Rahul Thanvi[1], Gaurav Mishra[2], Yogesh Govari[3] ,Vrushal Bagwe[4], Prof. Nidhi Chitalia[5]

[1]*BE student, Department of Information Technology St. Francis Institute of Technology Mumbai, India*
[2]*BE student, Department of Information Technology St. Francis Institute of Technology Mumbai, India*
[3]*BE student, Department of Information Technology St. Francis Institute of Technology Mumbai, India*
[4]*BE student, Department of Information Technology St. Francis Institute of Technology Mumbai, India*
[5]*Professor, Department of Information Technology St. Francis Institute of Technology Mumbai, India*

---***---

Abstract - *In the face of evolving digital landscapes, the 'Visual Assist' project addresses the critical need for inclusivity through innovative usecases of machine learning and computer vision. This system empowers individuals with visual impairments by combining the advanced object detection capabilities of the YOLOv5 algorithm and the user-friendly audio generation of Google Text-to-Speech (gTTS). Recognizing the limitations imposed by visual impairments, Visual Assist aims to connect the information gap and foster independent engagement with the environment. Leveraging the real-time processing power and accuracy of YOLOv5, Visual Assist analyzes visual input from cameras, swiftly identifies objects within the user's surroundings, and conveys this information through clear and concise auditory descriptions generated by gTTS. This seamless translation of visual data into readily-understandable audio empowers users to perceive and navigate their surroundings independently and confidently.*

Deployable across various platforms, including smartphones, Visual Assist grants individuals with visual impairments greater autonomy in navigating not only public environments but also their personal environments. Real-world testing and user feedback play a essential role in refining the system to ensure it effectively meets the needs of users. The potential impact of Visual Assist extends beyond enhanced mobility; it cultivates a feeling of self-assurance and independence for individuals with partial visual impairments, empowering them to confidently venture into their surroundings and discover new experiences.

***Key Words*: Visual impairment; Assistive technology; Object Detection; Computer vision; Machine learning; Accessibility; Inclusivity**

1. INTRODUCTION

In the swiftly changing technological environment, the convergence of computer vision and machine learning has become instrumental in addressing real-world challenges. These cutting-edge fields leverage sophisticated algorithms and state-of-the-art technologies, influencing diverse sectors such as self-driving cars, healthcare, entertainment, and notably, accessibility. This study explores the profound significance of accessibility within this evolving landscape, representing the combination of computer vision, machine learning, and artificial intelligence to create solution that enable people with disabilities to feel empowered. These solutions aim to close the divide between limitations and possibilities, particularly focusing on the challenges faced in the domain of accessibility.

Within the realm of accessibility, the project identifies major challenges, including the digital divide influenced by socioeconomic factors, standardization and interoperability issues in assistive technologies, ethical considerations surrounding privacy and bias in AI, and the performance optimization challenge for real-time applications such as Visual Assist. The rationale behind this research endeavor stems from a deep commitment to improving the lives of individuals with visual impairments in a technologically dominated world. Embracing the principle that technology ought be a bridge instead being a barrier, the project utilizes advanced machine learning and computer vision, specifically the YOLOv5 object detection algorithm and gTTS library, to provide audio feedback for objects in the environment. The overarching goal is to redefine accessibility, positioning technology as a method that get past barrier and offers new opportunities, guided by a vision of empowerment and inclusivity for all individuals, regardless of visual limitations.

2. LITERATURE REVIEW

The described AI powered visual assistance and combined reading support present a novel system for individuals completely blind, utilizing the RP3 Model B+ for its cost-effectiveness and compact design. The system incorporates a camera, obstacle avoidance sensors, and advanced image processing algorithms for object detection. Additionally, it features an image-to-text converter using ultrasonic sensors for distance measurement. While the technology offers real-time feedback through an eSpeak speech synthesizer, it has limitations, such as potential inaccuracies in crowded environments due to reliance on sensors and limited computational capabilities of the Raspberry Pi for advanced deep learning algorithms. The system lacks sophisticated features like wet-floor and staircase detection, and its efficiency is impacted by the trade-off between speed and accuracy, particularly in detecting smaller or uniquely scaled objects using the SSDLite model. The overall functionality requires a physical module, and despite its advantages, the system currently lacks certain advanced capabilities.[1]

In [2], The architecture of T-YOLO, a type of YOLO-v5 tailored for tiny vehicle detection, can be delineated into three distinct sections. Firstly, the backbone comprises conventional convolutional neural network (CNN) operations and a streamlined forwarding mechanism. Subsequently, the neck integrates and amalgamates the different characteristics derived from diverse convolutional layers. Lastly, the head encompasses convolutional layers dedicated to bounding box regression and class predictions. The study conducted training sessions using multiple T-YOLO models on a proprietary dataset, juxtaposing their precision and recall metrics against those of YOLO-v5 models pretrained on the COCO dataset. Notably, the focus of this system is specific, concentrating solely on detecting vehicles within parking lots.

In [3],The proposed mobile application, "OpenCV Basics," serves as an educational tool designed to facilitate the teaching of fundamental Computer Vision concepts. By leveraging built-in OpenCV libraries, the application offers students hands-on experience and practical examples to supplement theoretical coursework. Unlike traditional approaches, the application does not rely on external datasets, utilizing instead the robust capabilities of OpenCV for object detection. While its primary aim is educational, offering insights into the main principles of computer vision, it does not address real-world challenges. Rather, it presents as a pedagogical aid, providing a controlled environment for learning and experimentation. The implementation's simplicity in user interaction, achieved through restricted modifiable parameters, may limit its flexibility. While this design choice enhances usability, it may hinder exploration of nuanced algorithm configurations. Consequently, users may encounter difficulties when attempting to tailor the application to specific Computer Vision tasks requiring intricate parameter adjustments. Thus, while "OpenCV Basics" offers a valuable educational resource, its reliance on predefined parameter settings may pose limitations in addressing more complex scenarios.

[4] The proposed system harnesses the strengths of both YOLO and Fast R-CNN models to achieve heightened accuracy in object detection tasks. Notably, YOLO exhibits a significant reduction in background errors compared to Fast R-CNN, laying a strong foundation for improved performance. By integrating YOLO to filter out background detections from Fast R-CNN results, the system achieves a substantial enhancement in overall accuracy. One of the key advantages of the system lies in its simplicity, as the model can be constructed easily and trained directly on full images. While Fast R-CNN contributes to a 2.3% improvement in performance when combined with YOLO, it encounters challenges in accurately localizing objects. However, the incorporation of YOLOv5 ensures real-time detection speed, while the refinement provided by Fast R-CNN enables precise localization and contextual understanding for each detected object. This synergistic approach capitalizes on the strengths of both models, resulting in a system that excels in both accuracy and efficiency.

[5] This study seeks to explain. the difficulty in object detection utilizing Convolutional Neural Networks (CNNs), with a particular focus on the YOLO algorithm developed by researchers. It systematically explores various iterations of the YOLO algorithm, addressing the limitations of each version in subsequent iterations. Beginning with an understanding of the fundamental principles. of CNNs, the paper delves into the specific components and terminology associated with this family of algorithms. It provides a detailed explanation of the types of layers utilized in each algorithm, providing understanding of how they function. and contributions to the overall architecture. A significant benefit highlighted in the paper is the efficacy of CNNs in expressing and extracting features from input data, owing to their robust mathematical framework. However, despite their effectiveness, CNNs are not without challenges. The paper discusses issues related to interpretability, stemming from the complex internal representations of CNNs. Furthermore, the computational demands of machine learning models pose obstacles to real-time deployment, particularly on resource-constrained devices. These considerations underscore the necessity for continuous research. and development to address overcome these obstacles and release the complete capabilities of CNN-based object detection systems.

3. PROPOSED SYSTEM

The proposed Object Detection system uses YOLOv5 algorithm with Google Text-to-Speech. The said system is implemented using pytorch library in Python. The model required for system was trained in Jupyter Notebook and was finally deployed using website developed in Flask Framework.

3.1 Dataset

The proposed system utilizes the COCO (Common Objects in Context) dataset, as it is extensively employed for tasks like identifying objects, segmentation, and captioning in the field of computer vision. It encompasses a diverse collection of images illustrating intricate scenes featuring multiple objects within different contexts. The collection of images is designed to capture the challenges associated with understanding objects within their surroundings. The collection of images consists of 80 object categories.

An additional 3000 images were collected from other datasets and labelled using RoboFlow,a service that offers resources for managing, annotating, and preprocessing image datasets for computer vision projects.

3.2 YOLO Architecture

The proposed solution involves the integration of YOLO, a state-of-the-art object detection algorithm, to recognize and localize objects within the user's environment.

Fig 1 shows the working of the General YOLO architecture

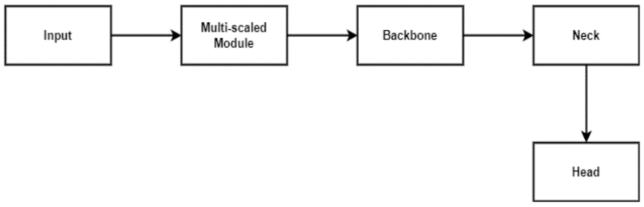


Fig -1: YOLOv5 General Architecture.

The structure of YOLOv5 comprises of three key components, each playing an essential role in the network's functionality:

1.Backbone (CSPDarknet):

The backbone of YOLOv5 is based on the CSPDarknet architecture, which stands for Cross Stage Partial (CSP) connections within the Darknet framework. CSPDarknet serves as the foundation of the network, in charge of taking features extracted from the input data. This procedure incorporates several levels of convolutions and pooling operations, allowing the network to progressively capture and abstract information from the input image.

2.Neck (PANet):

Following feature extraction by the backbone, the intermediate representations are passed through the PANet module. PANet, also known as Path Aggregation Network, enables the integration of features extracted from various levels of the network hierarchy. This fusion process enables the network to leverage both local and global context information, enhancing its ability to understand and interpret complex visual scenes.

3.Head (Yolo Layer):

The final stage of the YOLOv5 architecture is the Yolo Layer, where the detection results are generated. This component utilizes the fused features from the PANet module to predict bounding boxes, class probabilities, and confidence scores for the detected objects in the input image. The Yolo Layer applies a set of convolutional operations to produce these predictions, which include information about the class labels, confidence scores indicating the likelihood of detection, and precise spatial coordinates defining the location and dimensions of the identified objects.

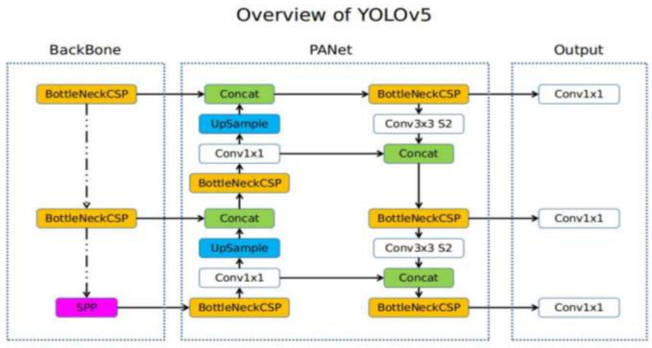


Fig -2: Yolov5 Architecture.[6]

In summary, YOLOv5's network architecture follows a hierarchical structure, with the Backbone responsible for feature extraction, the Neck for feature fusion, and the section dedicated to object detection and localization. This adaptable structure allows efficient processing of input data and robust detection of objects in various visual environments.

3.3 gTTS (Google Text-to-Speech)

gTTS (Google Text-to-Speech), a Python library used to interface with Google Translate's text-to-speech API. The library is used to write spoken mp3 data to a file, a file-like object (bytestring) for further audio manipulation. It features flexible pre-processing and tokenizing.

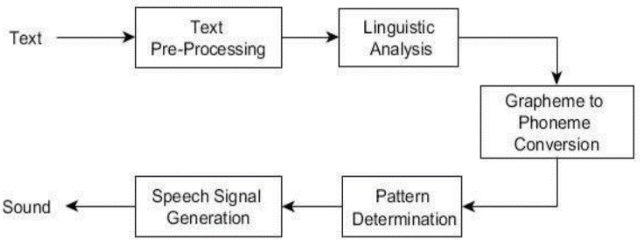


Fig -3: gTTS Architecture.

The system ingests textual input and employs natural language processing methodologies to comprehend its linguistic intricacies. Subsequently, it engages in logical inference on the parsed text. Following this, the text undergoes digital signal processing, where diverse algorithms and transformations are applied. Finally, the processed text is synthesized into speech format.

3.4 System Architecture

Fig 4 illustrates a flowchart outlining the training phase for the proposed system. The process commences with the collection of images from various datasets, which serves as the foundation to train the model in object detection. Subsequently, preprocessing is performed to prepare the images for further processing by labelling. Next, a suitable YOLO model is selected based on factors like accuracy, speed, and precision. The chosen model is then trained on the preprocessed image dataset. This training incurs significant computational resources and can take several hours or days to complete, depending on the dataset's size and model complexity. Following training, the model is tested on the validation set to refine the model's accuracy during training. Upon completion of training and tuning, the model's performance is evaluated on a fresh dataset of images to examine its accuracy and identify potential areas for improvement. Furthermore, hyperparameter tuning is employed to maximize the model's training process by adjusting its governing settings. The final model then identifies and detects items in the given input. Then, the GTTS library, responsible for text-to-speech conversion in Python, is integrated with the model to provide a speech output for the detected class.

The final outcome is delivered to the user instantly via the web application.

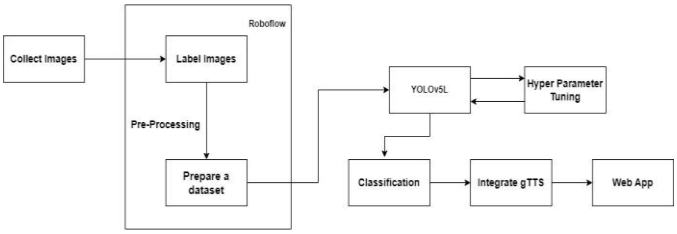


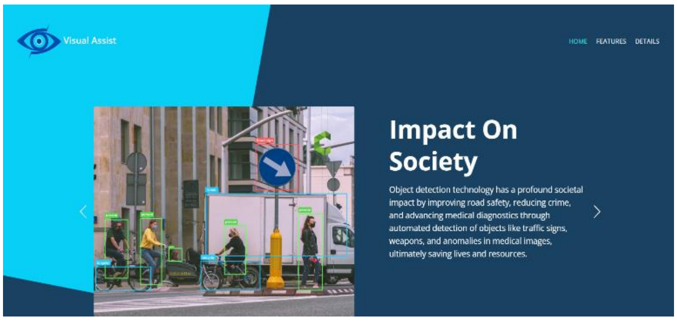
Fig -4: System Architecture.

4. IMPLEMENTATION

The outlined object detection system follows a systematic approach: it begins by accumulating a varied dataset using the collection of images provided by COCO and Roboflow for annotation. The dataset is carefully analyzed using Roboflow to examine confidence scores and overlapping bounding boxes, ensuring annotation quality. The YOLOv5 model is then trained using the collected dataset, with the YOLOv5L variant chosen based on computational resources and accuracy requirements. After successful training, the model is integrated with GTTS to deliver auditory responses regarding identified objects. in real-time. A web application is developed using Flask to serve as the user interface, allowing users to access the camera feed, view detected objects, and receive audio feedback seamlessly. The system is deployed using Ngrok as it provides secure tunnels to our local server, allowing external access without exposing our local machine to the internet.

5. RESULTS AND DISCUSSIONS

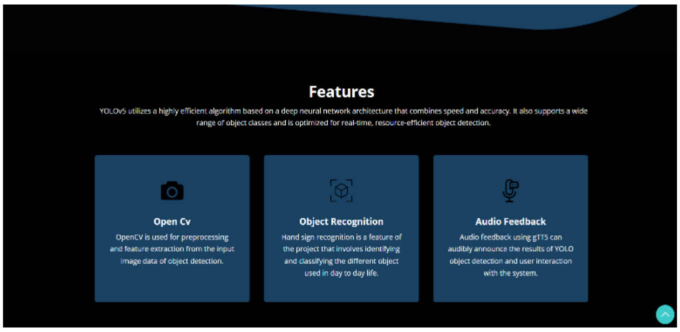
This work presents an machine learning model to identify objects achieving moderate to high accuracy, measured by mAP scores ranging from 49.0% to 67.3%. However, a crucial aspect to consider is the trade-off between accuracy and inference speed. While the model shows acceptable accuracy levels across both CPUs and V100 GPUs, its performance is significantly faster on the latter. This hardware dependency underscores the importance of carefully selecting hardware during deployment to ensure optimal performance. Furthermore, the model demonstrates potential for scalability, particularly on GPUs. Utilizing larger batch sizes (b32) on V100 GPUs leads to noticeably faster inference times compared to smaller batches (b1). This scalability potential suggests the model could handle increased workloads efficiently when properly configured.



Impact On Society

Object detection technology has a profound societal impact by improving road safety, reducing crime, and advancing medical diagnostics through automated detection of objects like traffic signs, weapons, and anomalies in medical images, ultimately saving lives and resources.

Fig -5 Web App Home Page



Features

YOLOv5 utilizes a highly efficient algorithm based on a deep neural network architecture that combines speed and accuracy. It also supports a wide range of object classes and is optimized for real-time, resource-efficient object detection.

Open Cv

OpenCV is used for preprocessing and feature extraction from the input image data of object detection.

Object Recognition

Hand sign recognition is a feature of the project that involves identifying and classifying the different object used in day to day life.

Audio Feedback

Audio feedback using gTTS can audibly announce the results of YOLO object detection and user interaction with the system.

Fig-6: Features Page

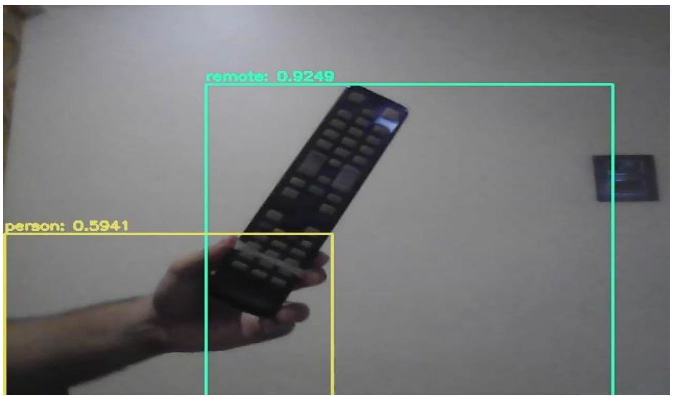


Fig-7: Multiple overlapping objects detected simultaneously

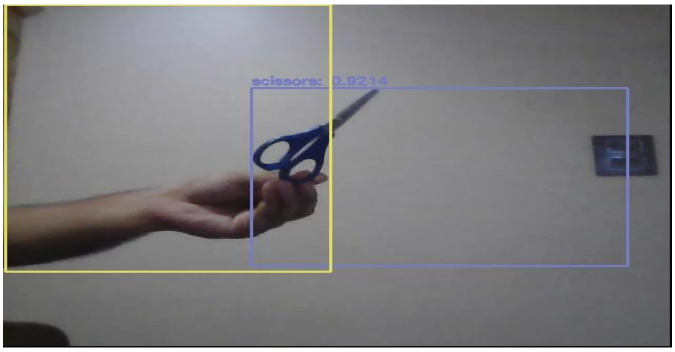


Fig-8: An individual scissor detected with a bounding box

Despite delivering moderate accuracy, the model's architecture remains relatively lightweight, boasting only 46.5 million parameters. Additionally, its computational complexity, measured in FLOPs, is moderate, demonstrating efficient resource utilization during inference. This efficiency contributes to the model's real-time capabilities, particularly evident on V100 GPUs. The ability to process frames rapidly makes it Appropriate for uses such as video monitoring or autonomous navigation systems, where timely response is crucial. However, opportunities for further optimization exist to enhance the model's performance across different hardware configurations while maintaining its accuracy. Exploring techniques like model pruning, quantization, or hardware-specific optimizations could potentially lead to improved inference speeds on a wider variety of platforms, making the model even more adaptable to various deployment scenarios.

Table -1: Performance Matrix

Performance Matrix		
YOLOv5L	mAP(50%)	mAP(50-95%)
COCO Dataset	0.67	0.49
Custom Dataset	0.63	0.479

5. CONCLUSION

In conclusion, the proposed system represents a significant breakthrough in accessibility technology, powered by the integration of YOLOv5 and gTTS. YOLOv5, a state-of-the-art object detection algorithm, serves as the backbone of the system, leveraging its exceptional accuracy and real-time processing capabilities to identify objects within the user's environment swiftly and accurately. The YOLOv5l model was trained on a combination of the COCO dataset and 3000 other images processed using Roboflow to obtain weights better suited for the proposed system.

Complementing YOLOv5's advanced object detection capabilities is the integration of gTTS, a powerful text-to-speech synthesis technology. Through gTTS, the proposed system converts detected objects into clear and concise auditory cues. This integration empowers users with visual impairments to delve into their surroundings confidently and independently, offering a seamless transition from visual to auditory perception.

The continued advancement and refinement of object detection models hold immense promise for further enhancing the proposed system's efficacy and usability. As technology continues to evolve, the proposed system stands poised to redefine the landscape of accessibility technology, empowering individuals with visual impairments to delve into their surroundings with confidence and independence.

REFERENCES

[1] M. A. Khan, P. Paul, M. Rashid, M. Hossain and M. A. R. Ahad, "An AI-Based Visual Aid With Incorporated Reading Assistant for the Fully Blind," in IEEE Transactions on Human-Machine Systems, vol. 50, no. 6, pp. 507-517, Dec. 2020, doi: 10.1109/THMS.2020.3027534.

[2] D. Padilla Carrasco, H. A. Rashwan, M. Á. García and D. Puig, "T-YOLO: Tiny Vehicle Detection Based on YOLO and Multi-Scale Convolutional Neural Networks," in IEEE Access, vol. 11, pp. 22430-22440, 2023, doi: 10.1109/ACCESS.2021.3137638.

[3] J. Sigut, M. Castro, R. Arnay and M. Sigut, "OpenCV Basics: Mobile Application to Support the Teaching of Computer Vision Concepts," in IEEE Transactions on Education, vol. 63, no. 4, pp. 328-335, Nov. 2020, doi: 10.1109/TE.2020.2993013.

[4] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, ”YOLO: Unified, Real Time Object Detection,” 2016 IEEE Conference on CVPR, Las Vegas, NV, USA, 2016, pp. 779-788, doi: 10.1109/CVPR.2016.91.

[5] J. Du, “Understanding of object detection based on CNN family and Yolo,” Journal of Physics: Conference Series, vol. 1004,p.012029,2018.doi:10.1088/17426596/1004/1/012029.

[6] Ultralytics, “Overview of model structure about Yolov5 ·issue280·ultralytics/yolov5,”GitHub,https://github.com/ultralytics/yolov5/issues/280 (accessed Aug. 29, 2023).

BIOGRAPHIES

Rahul Thanvi, Student, Dept. of Information Technology, St. Francis Institute of Technology, Mumbai, Maharashtra, India

Gaurav Mishra, Student, Dept. of Information Technology, St. Francis Institute of Technology, Mumbai, Maharashtra, India

Yogesh Govari, Student, Dept. of Information Technology, St. Francis Institute of Technology, Mumbai, Maharashtra, India

Vrushal Bagwe, Student, Dept. of Information Technology, St. Francis Institute of Technology, Mumbai, Maharashtra, India

Nidhi Chitalia, Asst. Professor, Dept. of Information Technology, St. Francis Institute of Technology, Mumbai, Maharashtra, India