ISSN: **3107-6513**

DATA SCIENCE TECHNIQUES TO HELP VISUALLY IMPAIRED USERS

Sahar F. Sabbeh[1], Ebtesam Ali[1], Jana Alqarni[1], Sara Alharbi[1]

[1] *College of Computer Science and Engineering, University of Jeddah, Jeddah 21493, Saudi Arabia*

[1] sfsabbeh@uj.edu.sa

***Abstract*— Visually impaired people face numerous difficulties in their basic daily life activities. technological interventions are used to help them to meet these challenges. However, the benefits of these technologies are not fully reachable for the visually impaired in the Arabic community. In this project we aim to use deep learning technology to help visually impaired users detect and recognize different objects. Additionally, the commands are given vocally as well as the output which enables users to better understand their surroundings. A dataset of objects gathered from daily scenes is created to apply the required recognition such as Chairs, Mobile phones, etc. The proposed method for the blind aims at expanding possibilities for them to achieve their full potential. Using You Only Look Once algorithm (YOLO) for object detection in a video stream. In addition, voice feedback using text-to-speech (TTS) in the Arabic language will be used to make it easier for the blind to get information about objects.**

***Keywords*— Visually Impaired, Deep Learning, Object Detection, YOLO, TTS**

I. INTRODUCTION

Vision impairment represents one of the most significant public health challenges worldwide, affecting approximately 1.3 billion people globally [1]. This condition extends beyond mere visual limitations, imposing substantial socioeconomic burdens on affected individuals and society at large. Visual impairment is frequently associated with difficulties in physical functioning, emotional distress, and reduced social participation, all of which collectively diminish quality of life [2]. The majority of visually impaired individuals reside in developing nations, including Saudi Arabia, where blindness affects 0.70%–6% of the population and visual impairment impacts an additional 7.8%–14% [2]. These statistics underscore the urgent need for accessible assistive technologies tailored to this population.

Visually impaired individuals encounter numerous daily challenges stemming from their inability to access visual information. Simple tasks such as understanding one's surroundings or locating specific objects become formidable obstacles. The act of identifying an object without physical contact is inherently difficult, and in certain situations, tactile exploration may pose safety risks. While traditional aids such as white canes, guide dogs, GPS devices, and Braille systems provide some degree of assistance, each presents inherent limitations. White canes may be lost or misplaced, guide dogs cannot navigate complex environments effectively, GPS devices excel at location tracking but fail at object identification, and Braille offers limited information accessibility and is restricted to users proficient in the system. These shortcomings perpetuate dependency and diminish the autonomy of visually impaired individuals.

Even with the recent rapid development of information technology, various conveniences for people have been provided. Nevertheless, there are still many inconveniences for visually impaired people, especially Arabic speakers. A considerable number of studies have been published to assist the visually impaired, and several technologies have been developed to aid them, yet none have provided real-time guidance for Arabic speakers in their daily activities. Finding what a blind person is looking for takes a considerable amount of time, and in some cases, physical contact with the object can be dangerous.

To address these challenges, this work proposes an innovative assistive system that leverages computer vision and artificial intelligence techniques to support visually impaired Arabic speakers in their daily routines. The system is designed to translate the visual world into comprehensible auditory information, enabling users to interact with their environment more independently. By integrating three core modules—object detection, speech-to-text, and text-to-speech—the system captures user commands, processes real-time visual data to identify requested objects, and delivers vocal feedback in Arabic. This approach aims to empower approximately 800,000 visually impaired Saudis [3] by providing them with a smarter, safer, and more autonomous daily living experience. The integration of Arabic-language voice recognition and guidance represents a critical advancement, as existing assistive technologies predominantly operate in English, creating a language barrier that limits accessibility for Arabic-speaking populations.

II. BACKGROUND

Over the past decade, numerous modern AI technologies have been developed to facilitate visually impaired people's lives and support their independence in many aspects, including but not limited to education, navigation, healthcare, and daily activities. Many organizations have focused their AI research and development in two main fields—assistive devices and mobile applications—resulting in tools that help the concerned group interact more efficiently with the world.

The development of mobile applications is a continuous process, as many needed features have emerged, such as recognizing the surrounding environment indoors and outdoors, color identification, reading electronic and handwritten texts, providing audio guidance to capture a printed page, saving

ISSN: 3107-6513

people's faces for recognition, estimating age, gender, and emotions, and recognizing currency notes. Besides mobile applications, many assistive devices have been enhanced by adding sensors, embedded systems, cameras, and the most recent upgrades integrated with AI. The majority of these devices are either portable or wearable.

The most prominent example of a portable device is the smart cane. Compared to the traditional cane, which provides only essential feedback about the immediate surroundings, the smart cane adds powerful capabilities. In 2016, the Indian Institute of Technology Delhi developed the Smart Cane device [4], a travel aid that uses ultrasonic technology to identify objects and obstacles around the person. It can detect objects above knee height—which a normal cane cannot—and combines this with audible feedback, helping users navigate with more confidence. Other smart canes are also paired with mobile applications to provide additional features, including finding the user's current location and exploring unfamiliar areas if the user becomes lost.

Additionally, smart glasses represent an important category of wearable assistive devices. Envision Assistive Technology company has introduced AI-powered Smart Glasses [5], which use artificial intelligence to organize different types of information from visual cues and verbally translate that information to the user. These glasses read documents aloud, identify acquaintances, find missing items in the house, and help the wearer use public transportation. Smart glasses can also read and translate digital and handwritten texts from various sources, including computer screens, posters, barcodes, timetables, and food packaging.

Despite these advancements, the majority of available technology solutions suffer from two critical shortcomings. First, these offered devices are very expensive, whereas visually impaired people predominantly belong to lower-income groups, making such technologies financially inaccessible for a large portion of the target population. Second, the capacities and services of these proposed systems are predominantly limited to English speakers. This linguistic restriction excludes a substantial segment of the visually impaired community who are native Arabic speakers and struggle to comprehend English-based auditory guidance.

Consequently, there is a clear and pressing need for a complete design of a framework that utilizes computer vision alongside Arabic text-to-speech and speech-to-text methods. Such a framework can effectively overcome the limitations of cost and language accessibility. By relying on affordable software-based techniques rather than costly proprietary hardware, and by delivering all vocal interactions in Arabic, this approach provides a dedicated, practical, and inclusive solution for Arabic-speaking visually impaired individuals, enabling them to understand their surrounding environment without the barrier of a foreign language.

The principal contribution of this work lies in the development of a low-cost, fully integrated assistive system that combines real-time object detection with Arabic speech-to-text and text-to-speech capabilities. Unlike existing commercial solutions that are prohibitively expensive and restricted to English, the proposed framework offers an affordable software-based alternative specifically tailored to Arabic-speaking visually impaired users. By enabling users to issue vocal commands in their native language and receive audible object identification responses in Arabic, this system addresses both the financial and linguistic barriers that have limited the adoption of assistive technologies in the Arabic world. This integration of computer vision with native-language voice interaction represents a practical step toward greater independence and improved quality of life for the visually impaired population in Saudi Arabia and beyond.

III. Related Work

Although numerous solutions have been proposed over the last decade, none offers a comprehensive system capable of assisting visually impaired individuals across all aspects of daily living. The following subsections present a selection of the work that has been carried out in this domain.

A low-cost handheld device named Viziyon was proposed in [6] to aid visually impaired users through computer vision. The device provides audio feedback regarding detected objects and their distance from the user. It employs deep learning with the Fast region-based convolutional neural network (Fast-RCNN) algorithm. The authors found that Fast-RCNN proved efficient for object detection compared to alternative mechanisms. In [7], two cameras were mounted on a blind person's glasses, and the system integrated computer vision, a GPS-free service, and ultrasonic sensors to deliver environmental information. The Speeded-Up Robust Features (SURF) descriptor was optimized for recognition, while sensors detected obstacles at medium to long ranges. In [8], a computer-vision-based system utilizing a multiple-camera network was built by placing cameras at various indoor locations. The user sends a request to locate a specific object, after which all cameras search from different perspectives. SURF was employed as the image detection algorithm for feature extraction from captured frames. In [9], an embedded system running on a Raspberry Pi was proposed to detect surrounding objects in real-time video streams, providing audio feedback regarding the identified items. The system utilized computer vision with R-CNN, SSD, and Caffe-tuned algorithms, trained on the CIFAR-10 dataset. In [10], a solution combining sensors, computer vision, and speech recognition was implemented to explore the environment and locate requested objects. The user initiates a scan via voice command, and the system responds with 3D audio playback while also calculating the distance between the object and the camera. The HoloToolkit was used to manage cameras, audio input, and 3D audio output. Convolutional Neural Networks (CNNs) along with YOLOv2 (You Only Look Once) were applied for object detection.

Several other studies have deployed computer vision models as mobile applications. In [11], a mobile application was designed to help visually impaired users recognize their surroundings through real-time object detection using the Single Shot Detector (SSD) algorithm and TensorFlow APIs. The

ISSN: 3107-6513

application features approximate distance calculation and generates voice-based wireless feedback indicating the object name and distance category (near, far, or medium). In [12], a mobile application was developed that allows users to recognize objects in a scene or video with voice feedback, read documents in English or Urdu, and detect and recognize currency notes. The YOLOv3 algorithm was chosen for detecting and recognizing both objects and currencies, described by the authors as one of the fastest and most accurate algorithms for such problems. In [13], the authors trained a computer vision model for Detecting and Identifying Objects and their Distances to aid Blind people (DIODB) through image-to-sound conversion. They utilized Scale Invariant Feature Transform (SIFT) and SURF algorithms for object detection and recognition, concluding that these methods offered robust matching, good processing speed, and resilience to variations in illumination, rotation, and scaling. In [14], a computer-vision-based prototype was proposed to provide walking assistance for visually impaired individuals by avoiding obstacles. Object detection was performed using YOLO, while depth estimation was achieved through monocular vision, eliminating the need for stereo camera setups. The findings indicated that algorithms such as YOLO and SSD outperformed standard CNN approaches for object detection and recognition. In [15], a deep learning model was trained for visually impaired users that operates by accepting a voice command indicating the desired object, then performing detection and recognition to guide the user. The authors found that CNN was accurate for classification but performed poorly in detection and localization. To address this, they employed Fast R-CNN, which utilizes linear regression, but encountered insufficient processing speed for real-time recognition. They also tested YOLO, though its accuracy was inadequate. Ultimately, they adopted SSD, as it does not generate candidate regions, trains faster than Faster R-CNN, and achieves higher accuracy than YOLO by leveraging multi-scale feature maps. In [16], a model was trained to detect objects in real-time video streams with high speed and provide audio output for detected objects in indoor environments. The implementation used the YOLO algorithm alongside OpenCV and the COCO dataset.

A comparative analysis was conducted between the proposed system and the reviewed academic literature. The findings reveal several important trends. Most studies employed YOLO, SSD, or SURF/SIFT as their primary object detection architectures, confirming that these models are the most commonly adopted for assistive vision applications. However, a critical gap exists in the area of language support and user interaction. Only the work reported in [7] offered Arabic language functionality, but this was limited solely to facial recognition rather than general object detection. Conversely, while studies such as [8], [10], [12], and [15] incorporated voice command capabilities to facilitate hands-free user interaction, none of these systems provided Arabic audio feedback or voice recognition. This disjointed approach highlights that no existing academic solution fully integrates both voice-based interaction and comprehensive Arabic-language support for general object detection within a single unified framework.

In addition to academic research, a review of widely used commercial mobile applications—including TapTapSee, RightHear, SeeingAI, and Supersense—was performed. These applications offer robust object detection and audio output functionalities, making them valuable tools for visually impaired users. However, all four applications lack built-in voice command features, requiring users to navigate through touch-based interfaces, which can be challenging for blind individuals. More critically, none of these commercial solutions support the Arabic language, severely limiting their accessibility for Arabic-speaking users who are unable to comprehend English-based audio guidance.

In contrast, the proposed system uniquely addresses these shortcomings by combining object detection, audio output, native Arabic text-to-speech and speech-to-text capabilities, and voice command functionality within a single, affordable framework. This sets the proposed work apart from both existing academic literature and commercial alternatives, as it represents the first fully integrated solution tailored specifically to the needs of Arabic-speaking visually impaired individuals, enabling them to interact with their environment independently using their native

IV. METHDOLOGY

The methodology adopted in this project follows the Data Science Life Cycle (DSLC), a structured framework that guides the development of data-driven AI systems from conception to deployment. The DSLC provides a systematic approach to building machine learning models by breaking the process into distinct phases: problem definition, data collection, data preparation, model training, model evaluation, and deployment. This structured methodology ensures that each stage is carefully executed, documented, and validated, ultimately leading to a robust and reliable object detection system tailored for visually impaired users.

In the context of this project, the DSLC begins with defining the core problem: developing a real-time object detection system that can identify indoor objects and provide Arabic audio feedback to visually impaired users. The subsequent phases involve collecting and annotating a custom dataset, preprocessing and augmenting the data to improve model generalization, selecting and training an appropriate object detection algorithm, evaluating the model's performance using relevant metrics, and finally deploying the system as a tool.

1. Data Collection

Object detection models require annotated datasets where each image is labeled with bounding boxes indicating the presence and location of objects. The annotation format must be compatible with the chosen detection algorithm. Three common formats are MS COCO, which stores annotations in a single JSON file [18]; PASCAL VOC, which uses individual XML files per image [18]; and YOLO, which employs a .txt file for each image containing the object class and normalized coordinates [19]. Based on our review of related literature, the majority of studies selected YOLO for real-time object detection due to its superior speed and accuracy, making it the most suitable choice for our assistive system [14]. Consequently, we adopted the YOLO annotation format for our dataset.

The data collection process is crucial for developing an efficient model. The quality and quantity of your dataset directly affect the AI model's decision-making process. These two factors determine the robustness, accuracy, and performance of the AI algorithms. As a result, collecting and structuring data is often more time-consuming than training the model on the data.

ISSN: **3107-6513**

There are different sources to acquire datasets: using a public image dataset or creating a custom dataset. In our project, we collected the data from a public source called Roboflow [20], in addition to using a web crawler to collect more images.

Roboflow is a Computer Vision developer framework for better data collection, pre-processing, and model training techniques. It has public datasets readily available to users and also allows users to upload their own custom data. It accepts various annotation formats. In data pre-processing, there are steps involved such as image orientations, resizing, contrasting, and data augmentations. One of the issues we faced during data collection was that there is no unified dataset containing all the classes we selected. So we had to download different datasets and merge them into a unified dataset. Also, after collecting the data, some classes were underrepresented. We are going to use different sources of data to collect more images, then add them into the dataset. This issue is explained in more detail in the data exploration section.

2. *Data description*

The goal of the dataset is to localize and detect eight indoor objects in images. For this reason, we have combined 8 datasets, each containing at least 100 images with its corresponding annotation file in YOLO format. The classes are as follows:

Mobile Phone – Chair – Water bottle – Forks – Sunglasses – Wallet – Medicine – Mugs

Annotations for each image are in form of a .txt file where each line of the text file describes a bounding box as shown in Fig.1.

FIG.1: YOLO annotation example

The annotation file for the image above is shown in Fig.2

zidane.txt

```
0  0.480109 0.631250 0.692969 0.713278
0  0.741016 0.522222 0.314844 0.933333
27 0.785937 0.506944 0.039062 0.151030
```

FIG.2: Yolo annotation file format

There are 3 objects in total (2 persons and one tie). Each line represents one of these objects. The specification for each line is as follows: a) One row per object, b) Each row is class x_center y_center width height format, c) Box coordinates must be normalized by the dimensions of the image (i.e., have values between 0 and 1), and d) Class numbers are zero-indexed (start from 0).

The dataset was collected with the objective to localize and detect eight indoor objects commonly encountered in daily life. Instead of using a single pre-existing dataset, we collected a unified dataset by combining multiple public sources available through Roboflow [20], supplemented with additional images collected via a web crawler to address class imbalances. The final dataset comprises a total of 6,057 images and 7,800 annotated instances distributed across the following eig**ht classes:**

Table 1. Dataset Description

Class	Images	Annotations	Resolution
Mug	249	279	416×416
Mobile	2,528	3,779	416×416
Chair	645	663	640×640
Water Bottle	168	168	416×416
Medicine	72	87	416×416
Wallet	1,320	1,553	640×640
Sunglasses	227	249	416×416
Fork	848	1,022	2048×1536
Total	**6,057**	**7,800**	—

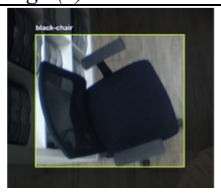
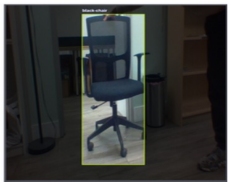
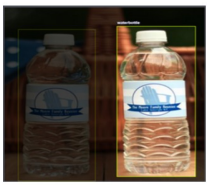
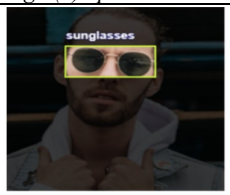
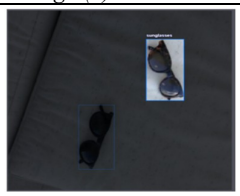
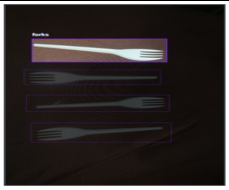
Each image in the dataset is accompanied by a corresponding annotation file in YOLO .txt format, where each line describes a bounding box with the object class and normalized coordinates [14]. All annotations were validated to ensure no missing labels or null examples, with the exception of the mug and medicine datasets which initially contained some missing annotations that were subsequently corrected during the preprocessing phase.

It can be observed that the dataset suffers class imbalance. The mobile phone class dominates with 2,528 images and 3,779 instances, while the medicine class contains only 72 images and 87 instances. This imbalance presents a challenge for model training, as the detector may become biased toward overrepresented classes. To mitigate this, data augmentation techniques and additional image collection through web crawling were employed to increase the representation of underrepresented classes, particularly medicine, water bottle, and sunglasses. Samples of the dataset are shown in *Table*.2.

The dataset also exhibits variation in image resolution across classes. Most classes are standardized at 416×416 pixels, which aligns well with the input requirements of the YOLO architecture [14]. However, the fork dataset contains images with a significantly higher resolution of 2048×1536 pixels, which required resizing during preprocessing to maintain consistency. Despite these variations, the unified dataset provides a sufficiently diverse and representative collection of indoor objects to train a robust object detection model for our assistive system. Sample images can be seen in Figures (3-9)

Table.2. Sample dataset images

ISSN: **3107-6513**

Object	Sample
Mug	  *Fig.3(a) one cup Fig3. (b) two Cups*
Mobile Phone	  *Fig.4(a)one mobile fig.4(b) two mobiles*
Chair	  *Fig.5(a) a chair at 90° Fig.5(b) a chair at a different angle*
Water Bottles	  *Fig.6(a) a water bottle Fig.6(b) two water bottles.*
Medicine	  *Fig.7(a)one medicine Fig.7(b) two medicines*
Wallet	  *Fig.7(a)Open Wallet Fig.7(b) closed Wallet*
Sunglasses	  *Fig.8(a)worn sunglasses Fig.8(b) unworn Sunglasses*
Forks	  *Fig.9(a)Broken fork Fig.9(b) Four Forks*

3. *Data Exploration*

Following the data collection phase, we conducted a thorough exploration of the merged dataset to gain insights into its composition, quality, and potential challenges. This exploration phase is essential for making informed decisions regarding data preprocessing, augmentation strategies, and model training parameters.

First, we explored the class distribution and as can be seen in Fig.10, the graph clearly shows the over-representation of the mobile phone class and the under-representation of classes such as medicine and water bottle.

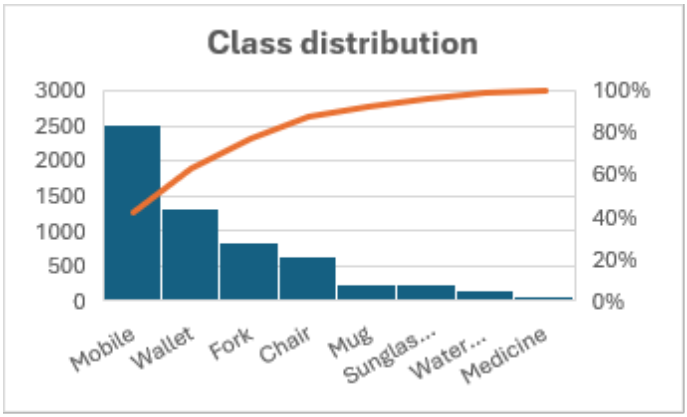
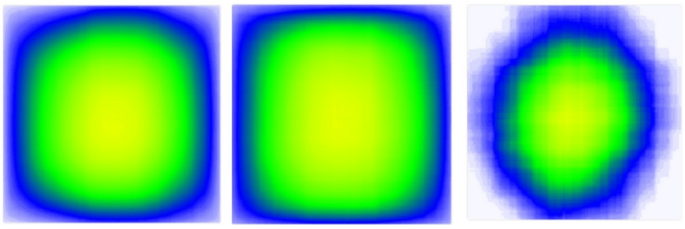


Fig.10. Class distribution

Secondly, annotation heatmap visualization was used to represent the spatial distribution of objects within images as shown in Fig.11. The merged heatmap for all classes reveals that annotations are mostly concentrated in the central region of the images. This pattern is particularly pronounced for the mobile phone and fork classes, which consistently appear near the image center. This central positioning is intuitive, as users typically center objects of interest when capturing photographs.



a)heatmap fo merged classes. b) heatmap for mobile c) heatmap for fork

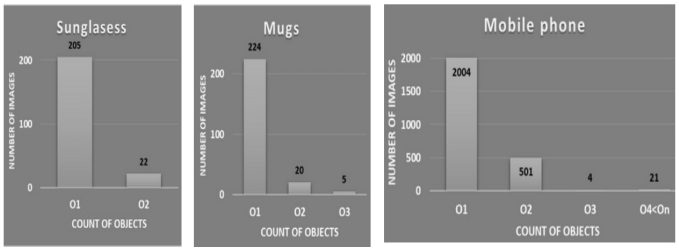
Fig. 11. Sample Annotation Heatmaps

Understanding the spatial distribution of annotations is valuable for model training, as it indicates that the model will likely perform better on centrally located objects. However, this also suggests that the model may struggle with objects positioned at the image peripheries. To address this limitation, we may consider data augmentation techniques that randomly crop or shift images to encourage the model to learn from objects appearing in various positions.

Afterwards, the object count histogram was used to explore the detailed information about the number of objects appearing in each image per class. For the sunglasses class, the histogram shows that images contain a maximum of two objects, with the majority containing exactly two sunglasses. The mugs class demonstrates slightly more variation, with images containing up

ISSN: 3107-6513

to three objects. Meanwhile, the mobile phone class exhibits the greatest variation, with images containing up to four or more objects, including a substantial number of images with three or four phones.



a)sunglasses Histogram. b) Mugs Histogram c) Mobile Histogram

Fig. 12. The Histogram of Count Objects

The data exploration process revealed several important characteristics of the used dataset:

1. **Class imbalance**: The dataset suffers from significant class imbalance, with mobile phones being heavily over-represented and several classes, particularly medicine and water bottle, being under-represented. This requires targeted augmentation and data collection strategies.
2. **Irrelevant/Outdated Images:** Old mobile phone models and unrelated images contaminated the dataset.
3. **Wrong Annotations:** Multiple datasets contained incorrectly placed bounding boxes or mislabeled objects.
4. **Image Resolution Variation:** Images ranged from 0.04 MP to 3.15 MP with varying aspect ratios.
5. **Limited View Diversity:** Mug images were captured only from the front angle
6. **Spatial distribution**: Objects are mostly centered in images, suggesting the need for augmentation techniques that improve generalization.
7. **Confusing Classes:** Some wallet images resembled mobile phones in shape and size.

4. Data Preprocessing

Based on the data exploration findings in the previous section, preprocessing decisions were made.

The first step in preprocessing is to understand the characteristics of the model that will be used in order to train it with data that to maximize its performance. As [21] referred that YOLO performs CNN-based object recognition in a box called an anchor, which means that the image size is reduced to a specific square size based on the version that is used (e.g., 640x640 in v5). Images in the collected datasets varied significantly in resolution, ranging from 0.04 MP to 3.15 MP with varying aspect ratios. YOLO requires a fixed input size (640×640 in v5) regardless of the original image size [21]. If images are not uniformly resized, serious distortion occurs on objects during the resizing process. That's why steps were made to a) resized all images to a uniform size of 640×640 pixels, b) maintain aspect ratio while resizing to avoid object distortion, and c) apply padding where necessary to preserve object proportions.

Afterwards, class imbalance problem had to be addressed as this imbalance can cause the model to become biased toward the majority class, resulting in poor detection accuracy for underrepresented objects. During this phase, new images were collected for mug, medicine, water bottle, and sunglasses. Old mobile phone images were replaced by new images.

Additionally, augmentation techniques were applied to increase sample diversity. Augmentation techniques applied are listed in Table. 3.

Table.3. Augmentation techniques

Augmentation Technique	Parameters
Horizontal Flip	Probability: 50%
Rotation	±15° to ±30°
Brightness Adjustment	±20%
Contrast Adjustment	±20%
Saturation Adjustment	±25%
Gaussian Blur	Kernel: 3×3
Random Crop	Scale: 80-100%

The augmentation and data collection efforts successfully balanced the dataset. The final class distribution is presented in Table.4.

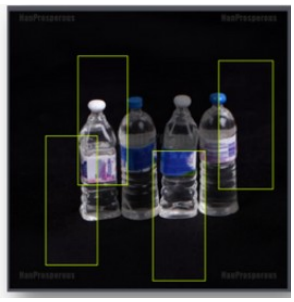
Table.4. Final Class distribution

Class	Before Preprocessing	After Preprocessing
Mobile Phone	3,779	1,494
Wallet	1,553	1,372
Fork	1,022	714
Chair	663	828
Mug	279	697
Sunglasses	249	722
Water Bottle	168	751

The next step was to address the problem of the incorrectly placed bounding boxes or mislabeled objects. These errors included bounding boxes that did not fully enclose the object, boxes that included background areas, or boxes that were positioned incorrectly. To solve this problem, manual review of all images was required to identify and fix the incorrectly placed bounding boxes. Samples are presented in Fig13.

ISSN: **3107-6513**

a)Medicine with wrong annotation



b)Bottle with wrong annotation

c)Wallet with wrong annotation

d)Forks with wrong annotation

Fig.13 Samples of incorrectly annotated images

Another problem was that some the mug images were notices to be all taken from the same viewpoint, thus new images were collected and to represent different perspectives (top-down, side, angled views). Other images were generated by applying rotation augmentation.

The class confusion problem appeared when some wallet images resembled mobile phones in shape and size, which could cause the model to misclassify wallets as mobile phones, reducing overall accuracy as shown in fig.14.

Fig.14. wallet image can be confused with mobile image

To solve this problem, wallet images that closely resembled mobile phones were identified, removed, and Ensured remaining wallet images had distinct visual features.

Finally, after completing all preprocessing steps, the final dataset was consolidated and analyzed. The following metadata summarizes the final dataset is shown in Figure. 15.

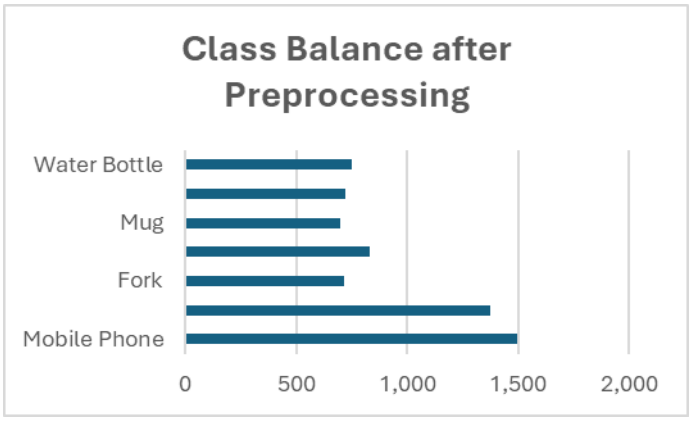


FIG.15: Final merge metadata

As shown, the dataset now contains **5,783 images**, and there are no missing annotations, meaning all image files have a matching annotation file. It contains **7,385 annotations** across the 5,783 images, approximately 1.3 per image, indicating good feature richness across all eight classes. The class distribution after preprocessing demonstrates a more balanced representation across all classes.

5. Model Building

This section describes the development of a real-time object detection system to assist visually impaired users. The work follows three stages: dataset preparation, model training, and system deployment. Since data collection was covered earlier, this section focuses on model selection, YOLOv5 training, and system integration.

Object detection combines two tasks: localization (finding where an object is) and classification (identifying what it is). The detector must predict a class label, confidence score, and bounding box for each object. In real-time video, inference speed and deployment constraints are as important as accuracy.

Deep-learning object detectors are commonly divided into two broad groups. **Two-stage detectors** first generate candidate regions and then classify those regions while refining their bounding boxes. This design can provide strong detection accuracy, but the additional region-proposal stage increases computational cost and makes the approach less suitable for streaming applications. The chapter identifies R-CNN, Faster R-CNN, and Mask R-CNN as representative examples of this family and retains the original citation to the source discussion [23].

One-stage detectors predict object classes and bounding boxes directly from the image without a separate proposal stage. This reduces the processing pipeline and makes the family attractive for real-time systems. YOLO, SSD, and RetinaNet are presented as examples. The project therefore selected the YOLO family, with YOLOv5 chosen because it combines an open-

ISSN: **3107-6513**

source implementation, a range of model sizes, PyTorch support, a relatively accessible production workflow, and compatibility with conversion to deployment-oriented formats.

YOLO predicts bounding boxes and class probabilities in a single forward pass over the full image. The approach is consequently well matched to a camera-based assistive system in which predictions must be generated continuously. The chapter's choice is also motivated by the availability of small models for resource-constrained devices and larger models for settings where accuracy is prioritized. The selected experiments compare YOLOv5s, YOLOv5m, and YOLOv5l rather than assuming in advance that the largest model will be the most appropriate for deployment.

YOLOv5 Detection Procedure and Architecture

The detection procedure described in the chapter has three operational elements. First, the input image is divided into a grid, and each cell predicts candidate bounding boxes together with confidence scores and class probabilities. Second, bounding-box regression estimates the center coordinates, width, and height of each box. Third, Intersection over Union (IoU) is used to quantify the overlap between predicted and reference boxes:

$$IoU = area\ of\ intersection\ /\ area\ of\ union$$

The resulting value lies between 0 and 1. A threshold can then be used to retain sufficiently overlapping predictions and suppress redundant or weak candidates. In the chapter's illustrative example, a threshold of 0.5 is discussed as a possible selection value. The final output is a set of non-redundant detections, each consisting of a class, confidence score, and bounding box.

The YOLOv5 configuration is organized into three principal components. The **backbone** extracts hierarchical visual features from the input image and is based on Cross Stage Partial (CSP) structures. The **neck** combines features at different scales; the chapter identifies PANet as the selected neck. The **head** converts the resulting features into detection outputs, including objectness scores, class probabilities, and bounding-box coordinates. Leaky ReLU is used in the intermediate layers, while the final detection layer uses a sigmoid activation as shown in fig.15.

The images were divided into training, validation, and testing subsets in a **70%/20%/10%** proportion. Google Colab supplied the cloud-based notebook environment and GPU acceleration used during training, while the Ultralytics YOLOv5 repository provided the implementation, model definitions, and configuration files.

The software environment included PyTorch and Torch vision for deep-learning, OpenCV and Pillow for image processing, NumPy and Pandas for data handling, Matplotlib and Seaborn for visualization, Tensor for monitoring, and packages such as PyYAML.

The training procedure used pretrained weights, an input resolution of 640 pixels, and a batch size of 32. 200 epochs were used for training and to extend training when early overfitting is not observed. Five experiments were made: YOLOv5s, YOLOv5m, and YOLOv5l trained for 200 epochs, followed by YOLOv5s and YOLOv5m trained for 500 epochs. The YOLOv5l model was not extended to 500 epochs because its 200-epoch run already required substantially more time than the smaller models.

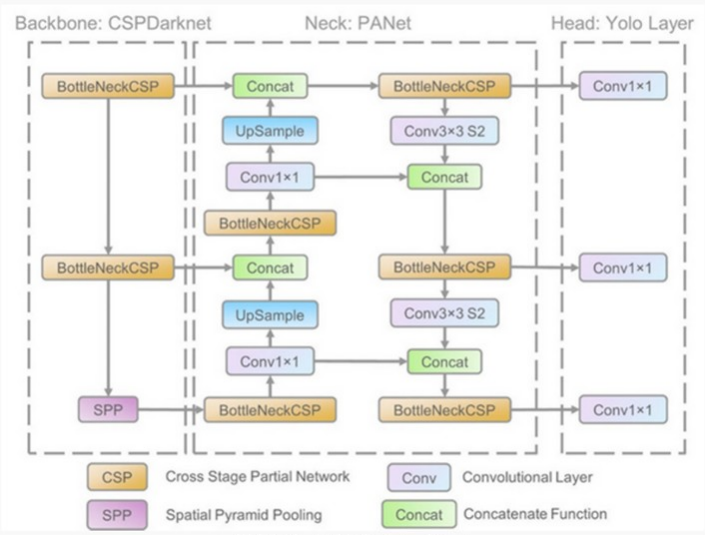


Figure 15. YOLOv5 architecture

The training-duration results show the expected computational effect of model size and epoch count. YOLOv5s and YOLOv5m required nearly identical times at both tested epoch counts, whereas YOLOv5l was considerably slower at 200 epochs. The results are summarized in Table 3.

Table 3. Training-duration comparison

Model	Epochs	Training duration
YOLOv5s	200	3 h 8 min
YOLOv5m	200	3 h 9 min
YOLOv5l	200	8 h 33 min
YOLOv5s	500	7 h 50 min
YOLOv5m	500	7 h 55 min

These measurements suggest that YOLOv5s and YOLOv5m were the most practical candidates under the available hardware constraints. The selection of the final deployment model, however, was appropriately left to the performance analysis described in the following chapter rather than being made from training time alone.

Performance Evaluation Metrics

To evaluate the performance of our object detection models, we employed several standard metrics commonly used in computer vision. These metrics provide a comprehensive

ISSN: 3107-6513

assessment of model accuracy, robustness, and suitability for real-world deployment [24].

- **Intersection Over Union (IOU)**

IOU measures the overlap between the predicted bounding box and the ground truth bounding box. The ratio ranges from 0.0 (no overlap) to 1.0 (exact match). In object detection evaluation, a detection is considered successful when the IOU exceeds a defined threshold. For instance, mAP50 represents accuracy when IOU ≥ 50%, while mAP75 requires a more stringent 75% overlap. Higher IOU thresholds indicate more precise localization requirements and are consequently more challenging to achieve [24].

- **Precision:**

Precision measures the model's ability to identify only relevant objects. It answers the question: "What proportion of positive identifications was actually correct?" A model with no false positives achieves a precision of 1.0. However, precision remains high even if some objects are missed, as it does not account for false negatives [24].

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (1)$$

- **Recall**

Recall measures the model's ability to find all ground truth objects. It answers the question: "What proportion of actual positives was identified correctly?" A model with no false negatives achieves a recall of 1.0. However, recall remains high even if many false positives are generated [24].

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (2)$$

- **Precision-Recall Curve:**

The precision-recall curve plots precision on the vertical axis against recall on the horizontal axis. This curve illustrates the tradeoff between the two metrics as the detection threshold varies. A higher threshold reduces false positives (increasing precision) but increases missed detections (decreasing recall). Conversely, a lower threshold detects more objects (increasing recall) but may generate more false positives (decreasing precision). A good object detection model maintains high precision even as recall increases, resulting in a curve that extends toward the upper right corner of the graph [24].

- **Average Precision (AP):**

AP represents the area under the precision-recall curve. A higher curve position toward the upper right corner yields a larger area, indicating better model performance. AP calculates the weighted mean of precision scores at each threshold, using the increase in recall from the previous threshold as the weight [24].

weight [24].

$$AP = \sum_{k=0}^{k=n-1} [\text{Recall}(k) - \text{Recall}(k+1)] \times \text{Precision}(k) \quad (3)$$

- **Mean Average Precision (mAP):**

mAP is the standard metric for evaluating object detection models. It is calculated as the average of AP values across all classes. This metric provides a single comprehensive score that reflects overall model performance across all categories [24].

$$mAP = \frac{1}{n} \sum_{k=1}^{k=n} AP_k \quad (4)$$

Where: AP_k is the AP of class k and n is the number of classes.

These metrics enable us to compare the performance of different YOLOv5 variants and select the optimal model for deployment in our assistive system [24].

V. EXPERIMENTAL RESULTS

In our study, we used Yolov5 (small , medium , and large) trained using 200 epochs and Yolo5 (small , and medium) trained using 500 epochs.

1) Yolov5s200 Experiments Results

The figure demonstrates the YOLOv5S_200 model's loss components, the first row represents the training box loss, objectness loss, and classification loss. The second row represents the validation box loss, objectness loss, and classification loss. These are indicators of how well an algorithm predicts an object as the epoch number is increasing.

We can see that the validation losses are very small for all the loss components, and they get smaller in each epoch, which means that the objects are accurately recognized during the training process.

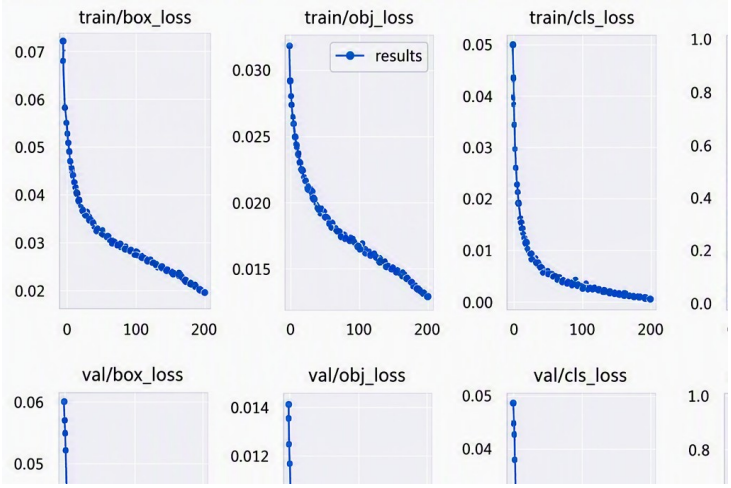


Fig.16: YOLOv5s200 Training loss

ISSN: **3107-6513**

Precision–Recall curve

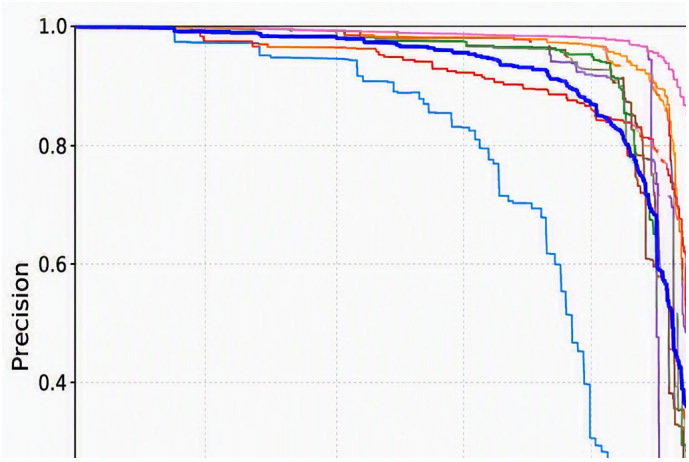


Fig.16: YOLOv5m200 Precision–Recall curve

Since the majority of the classes are at the higher upright of the graph, which made the AUC large, this means that the accuracy of the model is high. It can be seen that the model succeeded to identify sunglasses with 98% confidence and medicine comes next with approximately 96.6% confidence. Its noticed too that water bottles, mobile phones and cups were identified with approximately 94% confidence. Wallets and chairs with 93% confidence. It is also notices that the models worst performance was in identifying the forks with only 80% confidence.

The training results of the YOLOv5M_200 model can be observed in Table6 and the bar chart FIG17 below.

Table6: YOLOv5s200 table summary

Class	Precision	Recall	mAP@0.5	mAP@0.5 :0.95
All	0.922	0.873	0.929	0.655
Fork	0.827	0.71	0.801	0.363
Medicine	0.939	0.937	0.966	0.777
Mobile	0.959	0.87	0.94	0.692
Wallet	0.851	0.92	0.93	0.746
chair	0.976	0.874	0.935	0.67
Cups	0.931	0.894	0.942	0.702
Sunglasses	0.949	0.933	0.98	0.643
bottle	0.941	0.848	0.941	0.648

It's illustrated that the highest precision of the model was in identifying chairs with 98%. Afterwards, mobile phones 95.6%, sunglasses with 95.2%, water bottle with 94.9%, cups with 91.1% among all classes is Sunglasses and the lowest is Fork with 79%. Despite achieving the highest precision in chairs, its recall is only 83%. The highest recall was achieved in identifying sunglasses. Its noticed that the overall performance of the model in identifying the sunglasses almost outperforms its ability to identify all the other objects and this may be due its unique shape and visual features.

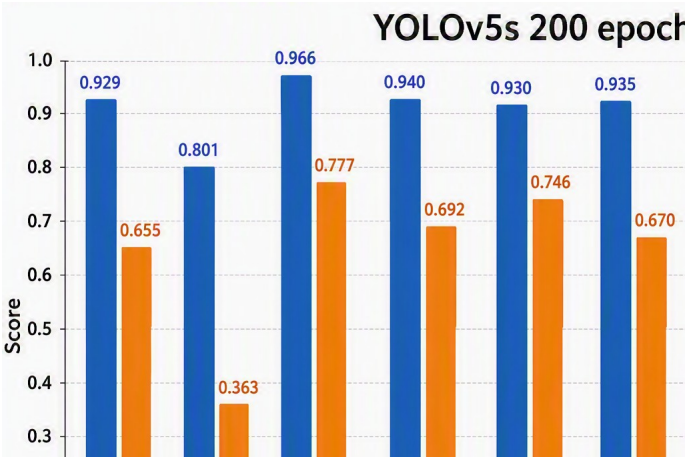


Fig.17: YOLOv5s200 summary

The overall mAP of all classes is 92.2% for Precision, 87.3% for Recall, and 92.9% mAP. Which is very good for the smallest model.

YOLOV5m 200 Experiments Results

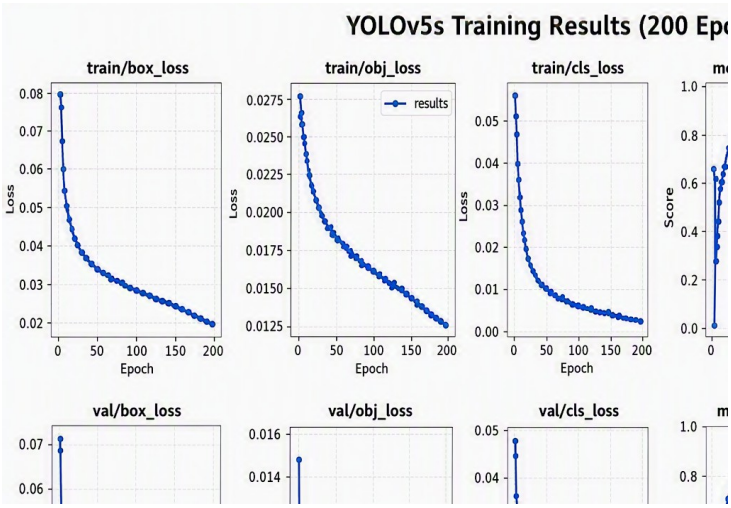


Fig.18:YOLOv5m200 training loss/ precision

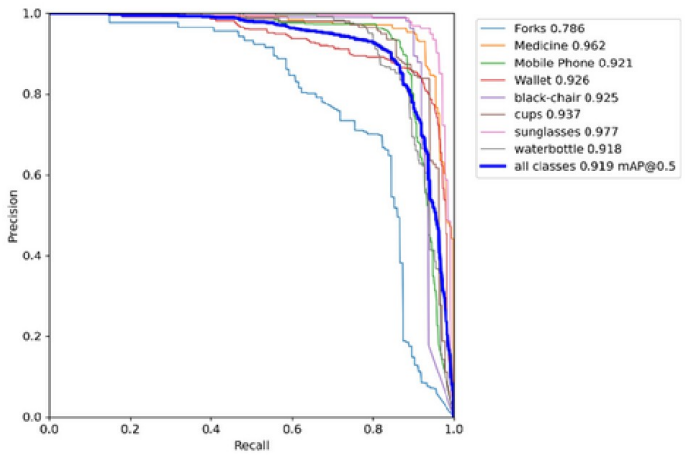


FIG19: YOLOv5m200 Precision–Recall curve

ISSN: 3107-6513

It's obvious that the performance is very similar to the YOLOv5s200. It can be seen that Yolo5m_200 has almost similar performance such as YOLOs_200 with only small difference in favor of YOLO5s_200. The model succeeded in identifying sunglasses with 97.7% confidence and medicine comes next with approximately 96.2% confidence. Cups come next with 93% confidence. Wallets, Chairs, mobile phones, and water bottles were identified with approximately 92% confidence. And again, forks identification was the lowest with 78.6% confidence.

Both YOLOv5s_200 YOLOv5M_200 missed detecting 1% of the Medicine class and detected it as a Water bottle, it also missed detecting 1% Water bottle class and detected it as a Medicine.

The training results of the YOLOv5M_200 model can be observed in Table.7 and the bar chart Fig.20 below. It's illustrated below that the highest precision of the model was in identifying chairs with 98%. Afterwards, mobile phones 95.6%, sunglasses with95.2%, water bottle with 94.9%, cups with 91.1% among all classes is Sunglasses and the lowest is Fork with 79%. Despite achieving the highest precision in chairs, its recall is only 83%. The highest recall was achieved in identifying sunglasses. Its noticed that the overall performance of the model in identifying the sunglasses almost outperforms its ability to identify all the other objects and this may be due its unique shape and visual features.

Table7.YOLOv5m200 table summary

Class	Precision	Recall	mAP@0.5	mAP@0.5: 0.95
All	0.911	0.849	0.919	0.639
Fork	0.799	0.647	0.786	0.364
Medicine	0.891	0.929	0.962	0.769
Mobile	0.956	0.835	0.921	0.647
Wallet	0.848	0.911	0.926	0.738
Chair	0.98	0.887	0.925	0.664
Cups	0.911	0.856	0.937	0.679
Sunglasses	0.952	0.94	0.977	0.638
Bottle	0.949	0.789	0.918	0.609

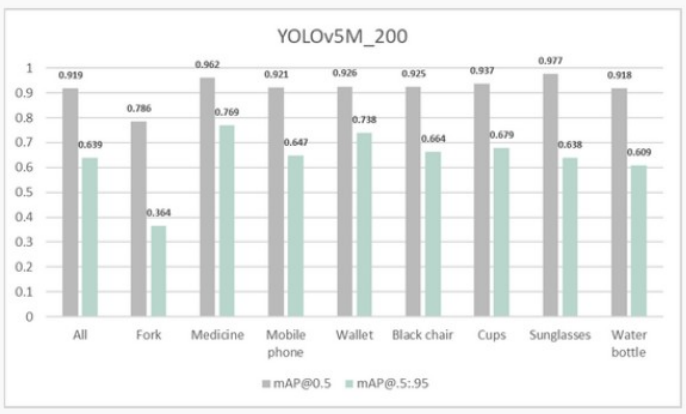


Fig20: YOLOv5m200 barchart summary

The model performed very well overall. It correctly identified objects about 92% of the time, which is excellent for helping visually impaired users find everyday items.

The system is especially good at detecting sunglasses, medicine, and cups, with accuracy above 93%. These objects are easy for the model to recognize because they have clear shapes and features.

The most challenging object was the fork, which the model detected correctly only about 79% of the time. This is because forks are thin and can be hard to see clearly, especially when they are mixed with other objects or in poor lighting. Water bottles also performed slightly lower than other classes but still achieved good results.

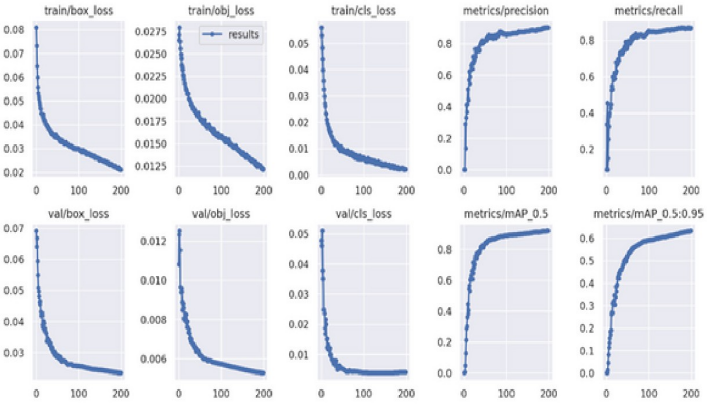


Fig.21.YOLOv5l training loss/precision

We can see that the class loses in the validation tend to be stable after 100 epochs

Precision–Recall curve.

As shown in Fig.22, training Yolo5L-200 achieved the same performance as its predecessors having sunglasses the highest with 97.3% confidence and forks at last with 77.6% confidence

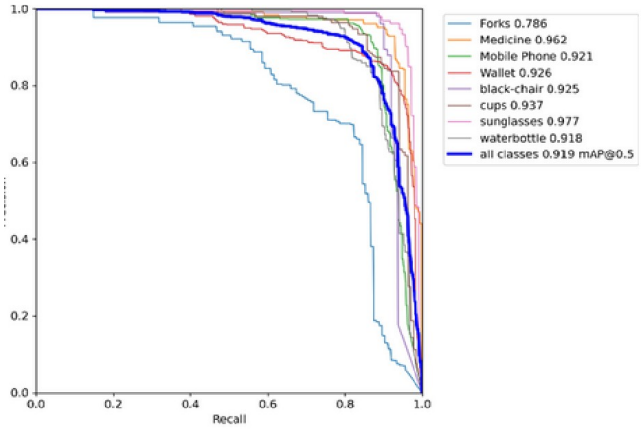


Fig.22.YOLOv5L the Precision–Recall curve.

The training results of the YOLOv5L_200 model can be observed in Table 8 and the bar chart Fig.23 below. The overall values of all classes have been calculated as 90.1% for Precision, 86.8% for Recall, and 92.0% mAP.

Table 8. YOLOv5l table summary

Class	Precision		Recall	mAP@0.5	mAP@0.5 :0.95
All	0.901		0.868	0.92	0.633
Fork	0.812		0.673	0.776	0.363
Medicine	0.884		0.929	0.96	0.759
Mobile	0.954		0.873	0.941	0.659
Wallet	0.844		0.923	0.933	0.738
Chair	0.968		0.892	0.928	0.653
Cups	0.865		0.864	0.919	0.648
Sunglasses	0.951		0.942	0.973	0.622

ISSN: **3107-6513**

Bottle	0.93		0.847	0.929	0.627

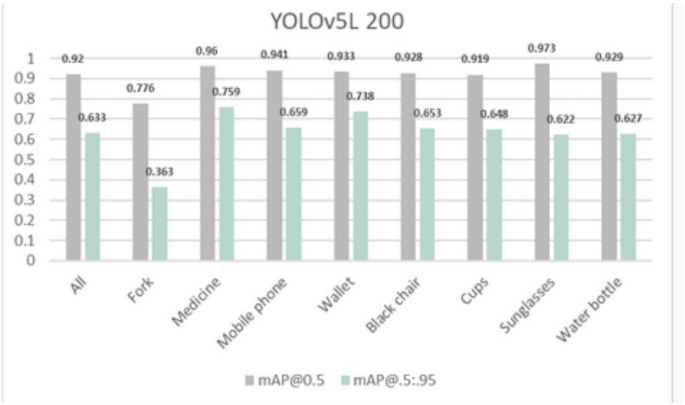


FIG23: YOLOv5l barchart summary

The model achieved strong overall performance with a mean average precision (mAP@0.5) of 92.0%, demonstrating reliable detection across all eight classes. The sunglasses class performed best with 97.3% accuracy, followed closely by medicine at 96.0% and mobile phones at 94.1%. The fork class remained the most challenging with only 77.6% accuracy, likely due to its thin and elongated shape which makes detection difficult in cluttered scenes.

Overall precision was 90.1% and recall was 86.8%, indicating the model makes few false detections while successfully finding most objects.

YOLOV5S 500 Experiments Results

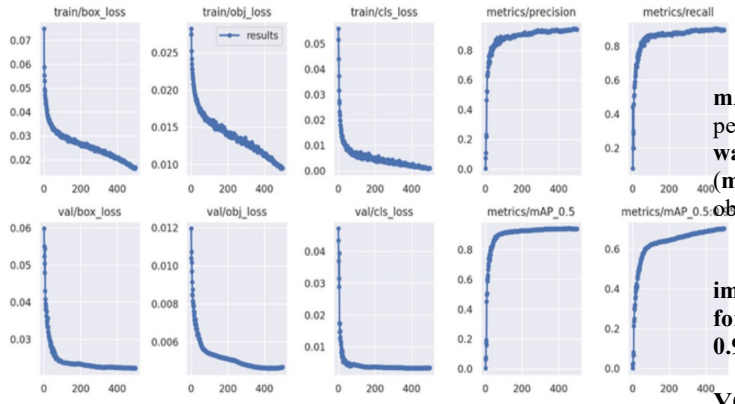


Fig.24.YOLOv5s500 training loss/precision

Precision–Recall curve.

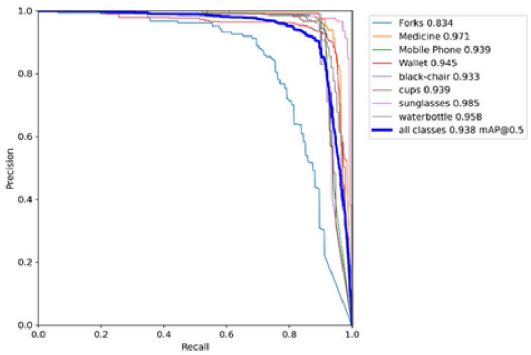


Fig.25: YOLOv5s500 the Precision–Recall curve.

The training results of the YOLOv5S_500 model can be observed in Table.9 and the bar chart FIG.25 below. The overall values of all classes have been calculated as 94.0% for Precision, 89.5% for Recall, and 93.8% mAp.

Table.9: YOLOv5s500 table summary

Class	Precision	Recall	mAP@0.5	mAP@0.5 :0.95
All	0.94	0.895	0.938	0.702
Fork	0.854	0.738	0.834	0.422
Medicine	0.933	0.939	0.971	0.809
Mobile	0.968	0.9	0.939	0.739
Wallet	0.891	0.939	0.945	0.792
Chair	0.989	0.901	0.933	0.715
Cups	0.956	0.902	0.939	0.73
Sunglasses	0.965	0.97	0.985	0.734
Bottle	0.966	0.871	0.958	0.671

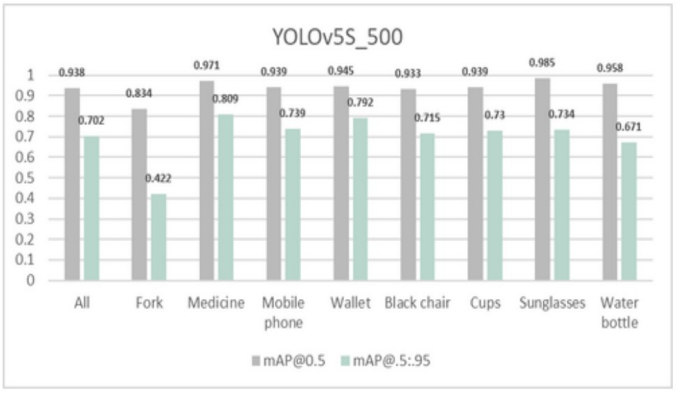


Fig.26 YOLOv5s500 barchart summary

The model achieved **excellent overall performance**, with an **mAP@0.5 of 0.938** and an **mAP@0.5:0.95 of 0.702**. The best-performing classes were **sunglasses (0.985)**, **medicine (0.971)**, and **water bottle (0.958)**. The **fork** class had the lowest results (**mAP@0.5 = 0.834** and **recall = 0.738**), making it the most difficult object to detect.

Compared with previous experiments, the model showed **clear improvements across all classes**. The largest gains were seen in the **fork** and **water bottle** classes. Overall, **mAP@0.5 increased from 0.920 to 0.938** and **mAP@0.5:0.95 increased from 0.633 to 0.702**.

YOLOV5M 500 Experiments Results

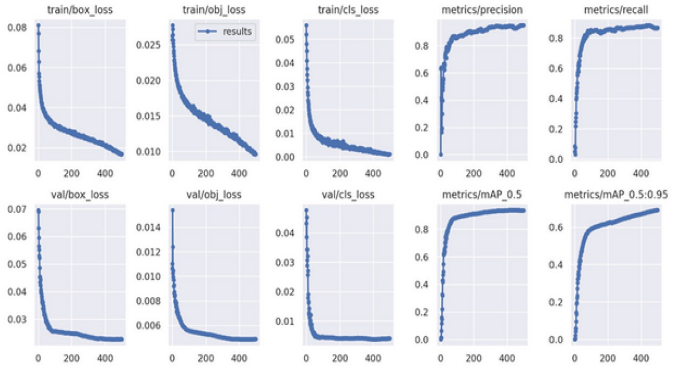


FIG27: YOLOv5m500 training loss/precision

ISSN: **3107-6513**

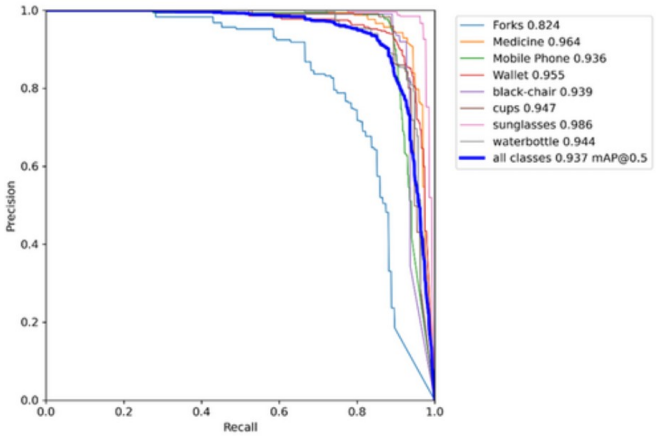


Fig.28: YOLOv5m500 the Precision–Recall curve.

The training results of the YOLOv5M_500 model can be observed in Table.10. and the fig.29.

Table.10: YOLOv5m500 table summary

Class	Precision	Recall	mAP@0.5	mAP@0.5 :0.95
All	0.95	0.866	0.937	0.691
Fork	0.889	0.667	0.824	0.408
Medicine	0.943	0.918	0.964	0.8
Mobile	0.978	0.87	0.936	0.729
Wallet	0.91	0.908	0.955	0.789
Chair	0.983	0.892	0.939	0.7
Cups	0.962	0.886	0.947	0.715
Sunglasses	0.982	0.963	0.986	0.718
Bottle	0.951	0.822	0.944	0.664

Results show that the model achieved excellent performance with an overall mAP@0.5 of **0.937**, demonstrating excellent detection accuracy at the standard 50% IOU threshold. The overall mAP@0.5:0.95 of **0.691** indicates strong localization precision.

Sunglasses and medicine show Exceptional performance, likely due to distinctive visual features making them easily distinguishable. Moreover, w**allet achieved** Moderate performance but balance between precision and recall. However, f**ork r**emains the most challenging class with the lowest recall, indicating that the model still struggles to consistently detect forks, likely due to their shape.

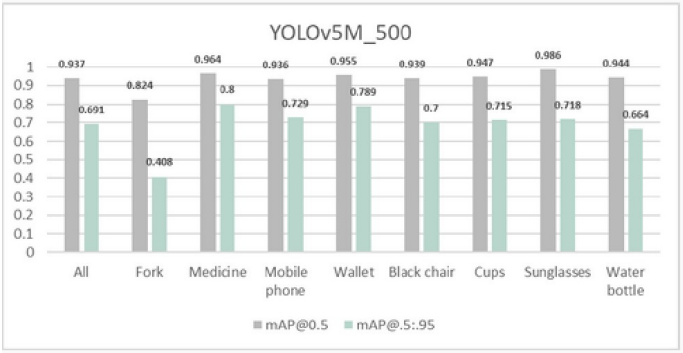


Fig.29: YOLOv5m500 barchart summary

VI. RESULTS DISCUSSION

The performance evaluation metrics was achieved in terms of precision, recall, mAP0.5, and mAP.5:.95. The evaluation was performed on different YOLOv5 models: YOLOv5s, YOLOv5m, and YOLOv5l. Table11 shows the comparison results.

Table.11: YOLOv5 models results comparison

Model	Precision	Recall	mAP@0.5	mAP@0.5 :0.95
YOLOv5s 200 epochs	0.922	0.873	0.929	0.655
YOLOv5m 200 epochs	0.911	0.849	0.919	0.639
YOLOv5l 200 epochs	0.901	0.868	0.92	0.633
YOLOv5s 500 epochs	0.94	0.895	0.938	0.702
YOLOv5m 500 epochs	0.95	0.866	0.937	0.691

Fig.30: YOLOv5 models barchart summary

According to Fig.30, YOLOv5S 500 model has the greatest mAP value. Also, we can conclude that the results are very close since the model was trained on the same architecture, and the difference was only in the parameter numbers. therefore, the results appeared very close between all models, and thus we can't determine which model is the best, consequently we will compare all models in the deployment section in real-time streaming to find out which model will detect the objects accurately.

Classes Comparison

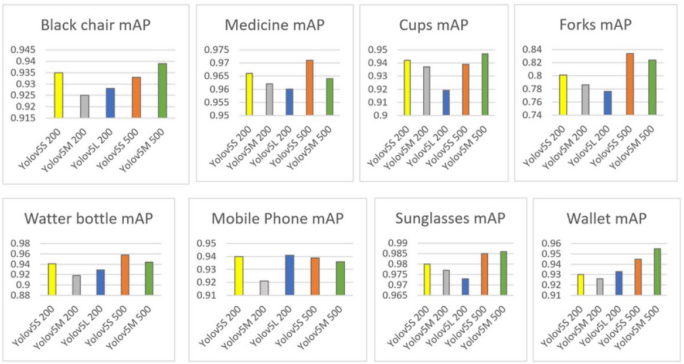


FIG6.31: Classes comparison

ISSN: 3107-6513

In Fig.31, we compared each class in each model to determine which one performs better, and based on the findings, we can see that YOLOv5s 500 performed better in detecting some objects such as Medicine, Water bottles, and Forks. While YOLOv5m 500 performed better in detecting other classes such as Wallet, Cups, and black chair. Therefore the comparison in real-time streaming is the best measure to determine which model to use.

Conclusion & future work

VII. CONCLUSION & FUTUIRE WORK

In recent years, some solutions have been devised to help the blind or visually impaired in detecting objects in their environment, but they are not efficient. Our purpose was to develop a system that can detect customized classes and a comfortable system for the blind to detect their surrounding objects. the system has two main characteristics, the first property is to detect the surrounding objects and produce an Arabic vocal output of their names, and the second feature is to search for a specific object according to the user's request, where the user can enter the name of the desired object in the Arabic language then searching for the object and produce audio and alarm to notify the user that the object has been found, Our advanced system uses a camera to seize real-time images in front of the users. The model and feature extraction technique used here is YOLOv5 and its framework trades with object detection by choosing the entire image in a single instance, and splitting the image into grids, then predicting the bounding box coordinates and class probabilities for these boxes. The biggest advantage of using YOLOv5 is its excellent speed. Our system also uses Google API for speech generation. The proposed method can detect the images and generate speech accurately will make the visually impaired virtually visible also it innovatively uses text-to-speech technology which provides Arabic audio descriptions of their surroundings and helps them in self-confidence.

In the future work, we will try to use and evaluate different YOLO and tiny YOLO versions to get the best performance. Additionally, we will train the experimented models to a wider range of objects. We will try to use suitable algorithms to approximate distance between the user and the object. Add more feature such as text recognition and reading to help users read long text, books, letters, and magazines.

ACKNOWLEDGMENT

This work was funded by the University of Jeddah, Jeddah, Saudi Arabia, under grant No. (UJ-23-FR-48). Therefore, the authors thank the University of Jeddah for its technical and financial support.

REFERENCES

[1] "Blindness and vision impairment," World Health Organization, 2021. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment. Accessed: Sep. 2022.

[2] M. Alsaqr, "Barriers to Low Vision Services Among Optometrists in Saudi Arabia," *Open Ophthalmology Journal*, vol. 15, pp. 178–188, Oct. 2021.

[3] "Barriers to Low Vision Services Among Optometrists in Saudi Arabia," 2021. [Online]. Available: https://openophthalmologyjournal.com/VOLUME/15/PAGE/178/FULLTEXT/. Accessed: Sep. 18, 2022.

[4] "SmartCane Device," Indian Institute of Technology Delhi, Apr. 2016. [Online]. Available: https://assistech.iitd.ac.in/smartcane.php. Accessed: Oct. 6, 2022.

[5] "Envision Announces AI-Powered Smart Glasses For The Blind And Visually Impaired," Envision, Jul. 2021. [Online]. Available: https://www.letsenvision.com/blog/envision-announces-ai-powered-smart-glasses-for-the-blind-and-visually-impaired?lang=en. Accessed: Oct. 6, 2022.

[6] S. M, J. Joy, A. Kuriakose, B. M B, A. K. Babu, and M. Kunjumon, "VIZIYON: Assistive handheld device for visually challenged," *Procedia Computer Science*, vol. 171, pp. 2486–2492, 2020. doi: 10.1016/j.procs.2020.04.269.

[7] J. Zraqou, W. Alkhadour, and M. Siam, "Real-time objects recognition approach for assisting blind people," *International Journal of Current Engineering and Technology*, vol. 7, no. 1, pp. 2347–5161, Feb. 2017.

[8] C. Yi, R. W. Flores, R. Chincha, and Y. Tian, "Finding objects for assisting blind people," *Network Modeling Analysis in Health Informatics and Bioinformatics*, vol. 2, no. 2, pp. 71–79, Feb. 2013. doi: 10.1007/s13721-013-0026-x.

[9] Y. Wong, J. Lai, S. Ranjit, R. Syafeeza, and A. Hamid, "Convolutional Neural Network for Object Detection System for Blind People," *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, pp. 1–6, 2019.

[10] M. Eckert, M. Blex, and C. Friedrich, "Object Detection Featuring 3D Audio Localization for Microsoft HoloLens," *In Proc. 11th Int. Joint Conf. on Biomedical Engineering Systems and Technologies*, vol. 5, pp. 555–561, Jan. 2018.

[11] A. Birambole, P. Bhagat, B. Mhatre, and Prof. A. Abhyankar, "Blind Person Assistant: Object Detection," *International Journal for Research in Applied Science and Engineering Technology*, vol. 10, no. 3, pp. 1168–1172, Mar. 2022. doi: 10.22214/ijraset.2022.40850.

[12] R. Patil, R. Modi, A. Parandekar, and J. B. Deone, "Designing Mobile Application for visually impaired and blind persons," *SSRN Electronic Journal*, 2022. doi: 10.2139/ssrn.4108763.

[13] A. Tamimi, A. Abu Hammad, A. Abdalla, and O. Al-Allaf, "A System for the Detection and Identification of Objects and Their Distances to Aid Blind People," *Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV)*, pp. 98–104, 2019.

[14] N. Thakurdesai, A. Tripathi, D. Butani, and S. Sankhe, "Vision: A Deep Learning Approach to provide walking assistance to the visually impaired," *arXiv preprint arXiv:1911.08739*, 2019.

[15] R. Dewangan and D. S. Chaubey, "Object Detection System with Voice Output using Python," *International Journal for Research

ISSN: **3107-6513**

Trends and Innovation (IJRTI)*, vol. 6, no. 3, pp. 2456–3315, 2021.

16] M. Kadhim and B. Oleiwi, "Blind Assistive System based on Real Time Object Recognition using Machine learning," *Engineering and Technology Journal*, vol. 40, no. 1, pp. 159–165, Jan. 2022. doi: 10.30684/etj.v40i1.1933.

[17] V. Mane, "Indian Currency Note images dataset 2020," Kaggle. [Online]. Available: https://www.kaggle.com/datasets/vishalmane109/indian-currency-note-images-dataset-2020. Accessed: Oct. 2022.

[18] R. Khandelwal, "COCO data format for Object detection," Medium, Dec. 09, 2019. [Online]. Available: https://towardsdatascience.com/coco-data-format-for-object-detection-a4c5eaf518c5. Accessed: Oct. 22, 2022.

[19] S. Pokhrel, "Image Data Labelling and Annotation—Everything you need to know," Medium, Mar. 11, 2020. [Online]. Available: https://towardsdatascience.com/image-data-labelling-and-annotation-everything-you-need-to-know-86ede6c684b1. Accessed: Oct. 22, 2022.

[20] "Roboflow: Give your software the power to see objects in images and video," Roboflow. [Online]. Available: https://roboflow.com/. Accessed: Oct. 19, 2022.

[21] "Image Preprocessing for Efficient Training of YOLO Deep Learning Networks," IEEE Conference Publication, IEEE Xplore. [Online]. Available: https://ieeexplore.ieee.org/document/8367193. Accessed: Nov. 2, 2022.

[22] Ultralytics, "GitHub - ultralytics/yolov5: YOLOv5 in PyTorch > ONNX > CoreML > TFLite," GitHub, Nov. 1, 2022. [Online]. Available: https://github.com/ultralytics/yolov5. Accessed: Nov. 2, 2022.

[23] S. Ansari, *Building Computer Vision Applications Using Artificial Neural Networks: With Step-By-Step Examples in OpenCV and TensorFlow with Python*, 2020. doi: 10.1007/978-1-4842-5887-3.

[24] D. Cochard, "mAP: Evaluation metric for object detection models," 2021. [Online]. Available: https://medium.com/axinc-ai/map-evaluation-metric-of-object-detection-model-dd20e2dc2472.