Real-Time Comprehensive Assistance for Visually Impaired Navigation

Amal Al-Shahrani [1] , Amjad Alghamdi[2], Areej Alqurashi[3], Raghad Alzahrani[4], Nuha imam[5]
amshahrani@uqu.edu.sa (corresponding author), s442004491@st.uqu.edu.sa, s442001962@st.uqu.edu.sa, s442017353@st.uqu.edu.sa, s441017954@st.uqu.edu.sa
University of Umm Al-Qura, College of Computing, Mecca, KSA

Abstract
Individuals with visual impairments face numerous challenges in their daily lives, with navigating streets and public spaces being particularly daunting. The inability to identify safe crossing locations and assess the feasibility of crossing significantly restricts their mobility and independence. Globally, an estimated 285 million people suffer from visual impairment, with 39 million categorized as blind and 246 million as visually impaired, according to the World Health Organization. In Saudi Arabia alone, there are approximately 159 thousand blind individuals, as per unofficial statistics. The profound impact of visual impairments on daily activities underscores the urgent need for solutions to improve mobility and enhance safety. This study aims to address this pressing issue by leveraging computer vision and deep learning techniques to enhance object detection capabilities. Two models were trained to detect objects: one focused on street crossing obstacles, and the other aimed to search for objects. The first model was trained on a dataset comprising 5283 images of road obstacles and traffic signals, annotated to create a labeled dataset. Subsequently, it was trained using the YOLOv8 and YOLOv5 models, with YOLOv5 achieving a satisfactory accuracy of 84%. The second model was trained on the COCO dataset using YOLOv5, yielding an impressive accuracy of 94%. By improving object detection capabilities through advanced technology, this research seeks to empower individuals with visual impairments, enhancing their mobility, independence, and overall quality of life.

Keywords:
Visual impairment, Object detection, Machine learning, Computer vision, Voice command functionality,
You-Only-Look-Once (YOLO).

1. Introduction

In recent years, there has been ongoing research and development in the field of applications and digital devices aimed at serving people with visual impairments. In the research that was done in 2021 by Senjam, S., et al. [11], researchers mentioned that smartphones have gained wide acceptance and are now less stigmatized compared to traditional assistive devices. Additionally, the number of apps specifically designed for people with visual impairments is rapidly increasing. For example, there are applications like VoiceOver, Aipoly Vision, TapTapSee, Be My Eyes, Seeing AI, and Seeing Assistant Move. However, it is worth noting that most of these apps are not considered safe or designed specifically to serve people with visual impairments in the process of moving around and crossing roads safely. Similarly,

In 2022, Mehmood et al. [16] aimed to understand the requirements and challenges faced by blind and visually impaired people in the Kingdom of Saudi Arabia regarding the availability and use of digital devices and applications. To achieve this, an online survey was conducted using digital forms, in which 164 participants participated. Participants reported using White Cane, mobile phones, Envision, Seeing AI, VoiceOver, and Google Maps. Mobility emerged as the most common purpose for using private devices among participants. Moreover, white canes and mobile phones were the basic tools used by the visually impaired, at a rate of 49% and 84% of respondents, respectively.

Then, in 2009, Jinqiang Bai et al. [1] discussed the use of deep learning machines, SLAM algorithms, and OCR algorithms in guiding blind people in unfamiliar environments. Also, in 2014, Krizhevsky et al. [2] proposed the R-CNN algorithm used for detection objects and implemented it on the ILSVRC-2010 dataset, achieving an error rate of 39.8% and a mAP of 60%. In 2015, Girshick et al. [3] introduced SPPNet, which was more than 20 times faster than R-CNN while maintaining the same detection accuracy (VOC07 mAP~=59.2%). Although SPPNet improved the detection speed effectively, it still had some drawbacks, such as a multi-stage training process. Girshick et al. [3] addressed the issues of previous methods, and Fast R-CNN demonstrated significant improvements in training and testing speed. It trained the VGG16 network much faster than both R-CNN and SPPnet, achieving a training speed that was 9 times faster than R-CNN and 3 times faster than SPPnet. At test time, Fast R-CNN was 213 times faster than R-CNN and 10 times faster than SPPnet. Additionally, Fast R-CNN achieved a higher mean average precision (mAP) on the PASCAL VOC 2012 dataset compared to R-CNN and SPPnet.

There was a review conducted in 2015 by Caldini et al. [4]. They were talking about an augmented electronic travel (ETA) system based on smartphones. By utilizing the smartphone's camera and gyroscope sensor, the

Manuscript received May 5, 2024
Manuscript revised May 20, 2024
https://doi.org/10.22937/IJCSNS.2024.24.5.1

structure-from-motion (SfM) algorithm calculates scene depth and estimates rotation between images. Gyro sensor data reduces drift errors, so the algorithm generates a 3D map by comparing points, estimating the principal matrix, and triangulating points. On the other hand, in 2016, Khenkar et al. [5] introduced an assistive system. It combines GPS technology and a novel obstacle detection method to make intelligent navigation decisions. The research used a dataset of around 52,000 classified images, categorized as obstacles and non-obstacles. A supervised machine learning approach using decision trees generates prediction models.

In addition, in 2017, Coughlan J. et al. [6] developed a system to assist individuals with visual impairments. The system was tested on blind volunteers in various indoor and outdoor environments. Moreover, in 2018, Ghilardi et al. [7] focused their research on assisting blind individuals in safely crossing the road. The study develops a system that utilizes a dataset called PTLD, which comprises 4,399 classified images of pedestrian traffic categorized into "GO," "STOP," and "OFF" states. The study evaluates the performance of different object detection models, including Faster R-CNN, YOLO Full, YOLO Tiny, and SSD. The SSD model achieves the highest average mean precision (mAP) and average precision (AP) across all classes, indicating its superior performance in accurately detecting pedestrians and their states, contributing to safer road crossing for blind individuals. Then, in 2019, according to Abdul Muhsin M. et al. [8], the research aims to assist visually impaired individuals in navigating and identifying objects in their surroundings using the YOLO (You Only Look Once) object detection algorithm on a Raspberry Pi 3 using Python and OpenCV. Likewise, in 2021, based on the research that was done by Miss Rajeshvaree et al. [9], their study introduces an object detection system that uses the YOLO algorithm and text-to-speech technology to detect objects by using a camera. It uses a COCO data set, and the system provides audio announcements to the blind about the object locations and image. The results show that it processes 45 frames per second, so it's incredibly fast and also efficient because it predicts more than one object from a single image.

However, in 2021, Montezuma et al. [10] focused on testing and comparing the results between two devices, Orcam MyEye 1 and Seeing AI, to ensure that the two applications accurately perform tasks and serve people with visual impairment. Both applications achieved over 95% accuracy for plain text documents; however, accuracy dropped to a range of 13% to 57% for text formatted on curved surfaces. Participants successfully completed 71% and 55% of tasks using Orcam MyEye 1 and Seeing AI, respectively. In a similar vein, in 2021, Salunkhe et al. [12] developed an Android-based object recognition application that utilizes the smartphone's camera to capture real-time images, which are processed using TensorFlow's object detection API, specifically the SSD algorithm. Detected objects are then converted into audio output through Android's text-to-speech library. The system achieved an overall accuracy of around 90% based on experimental evaluations.

There is also research; in 2022, See A et al. [13] proposed a system that integrates obstacle detection and object detection into a single application. The paper utilizes a deep learning model and the YOLO v3 framework for multi-object detection. The obstacle detection is based on the ARCore Depth Lab API by Google, which generates a 3D depth map using a depth-from-motion algorithm. determines the obstacle's location and provides audio warnings. The object detection feature utilizes the TensorFlow Lite framework and a trained model called COCO SSD MobileNet v2. The database enables over 90 different classes of objects; this model can detect objects from a custom training database and identify their categories. Furthermore, in 2022, Patil et al. [14] discussed a mobile application that integrates multiple functions. The application utilizes artificial intelligence and machine learning techniques, including the YOLOv3 algorithm for object recognition, which is known for its speed and accuracy. Additionally, the application incorporates a currency recognition model using TensorFlow and a dataset from Kaggle, consisting of over 1,000 different objects.

Furthermore, in 2022, Aniket Birambole et al. [15] proposed the Single Shot Multibox Detector (SSD) algorithm to detect objects in real-time. The experimental setup includes the use of TensorFlow, a deep learning library from Google, and training the neural networks on a dataset of images. The results demonstrate improved accuracy and efficiency in object detection compared to previous approaches in the same domain. Additionally, in 2023, Kuriakose et al. [17] highlighted the limitations of current navigation assistants, specifically their lack of portability and comfort. To tackle this issue, the researchers developed an application called DeepNAVI. They conducted tests using custom datasets consisting of 20 different types of obstacles and 20 scene categories relevant to navigation. The obstacle detection and scene recognition tasks were performed using deep neural networks, specifically the YOLO and Faster R-CNN algorithms. Moreover, the results demonstrated an accuracy of 87.8% in obstacle detection and 85% in scene recognition. Furthermore, in 2023, Sarmah et al. [18], Proposed a system Object detection and conversion of text to speech systems for visually impaired individuals were developed using the YOLOv4 model for object detection and the gTTs module for text-to-speech conversion. The results showed high performance, with an average processing time of less than three minutes for A4 paper and an error rate of 2%.

Previous research in our field has highlighted a significant gap concerning the safe crossing of roads, including the real-time detection of obstacles and traffic signals. Additionally, there is a lack of integration of voice command functionality to assist visually impaired individuals in searching for specific objects in their immediate surroundings. By addressing these research gaps, we aim to empower visually impaired individuals to autonomously navigate public spaces, enhancing their safety and overall quality of life.

2. DATASET

A. COCO (Common Objects in Context)

Is a large-scale object detection dataset created by Microsoft Research in 2014; it consists of 80 labels and 330k images. Dataset having annotations for object detection, segmentation, and captioning tasks [19].

B. Street crossing and street obstacles dataset:

The street crossing dataset and the street obstacles dataset were both collected from various sources, including Roboflow [20] and the Google image search engine. These datasets were categorized. In the street crossing dataset, the categories included pedestrian lines that alert and guide visually impaired individuals of the presence of the street, vehicles and cars encountered during road crossings, and traffic signals with colors indicating stop, caution, and go. On the other hand, the street obstacles dataset encompassed potholes, vehicles, traffic cones, road barriers, and natural obstacles like branches and trees. Each category in both datasets comprised 500 images, meticulously selected to provide a diverse and balanced dataset for training our model.

1) Image Labeling Using RoboFlow:

We conducted the image labeling process manually using RoboFlow. We carefully assigned 10 classes to the images. As shown in Table 1, these classes were selected to distinguish various objects commonly encountered on the roads. Some of them are used for detecting obstacles, while others are related to crossing road. We then created annotations for each image by accurately identifying the objects depicted in them, as shown in Figures 1 and 2.

TABLE 1. THE DATASET CLASSES

CLASS	PURPOSE
Bicycle	detect obstacles / crossing the road
Barrier	detect obstacles
Crosswalk	crossing the road
Car	detect obstacles for crossing the road
Green traffic light	crossing the road
Red traffic light	crossing the road
Yellow traffic light	crossing the road
Traffic comes	detect obstacles
Potholes	detect obstacles
Tree	detect obstacles

Fig.1. some example of class before labeling

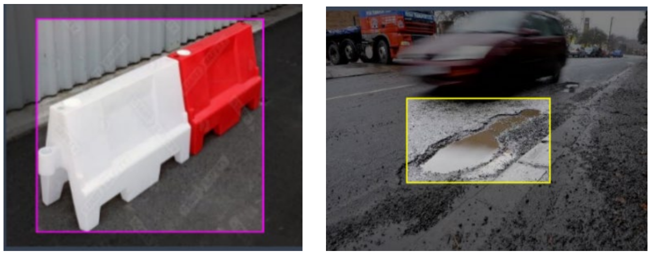


Fig.2. some example of classs aftar labeling

The final result after the labeling process is that the dataset consists of a total of 5283 images, which have been divided into three distinct sets: the training set, the validation set, and the test set. The training set comprises 4253 images, accounting for 81% of the dataset. The validation set contains 629 images, representing 12% of the dataset. Finally, the test set consists of 401 images, making up the remaining 8% of the dataset.

3. Methodology

A. YOLO Algorithms

YOLO (You Only Look Once) is an object detection algorithm. It is an innovative approach in the field of object detection, aiming to achieve fast and accurate detection of objects in images. Instead of dividing the image into small regions and analyzing them separately, YOLO divides the image into a grid of cells and performs object predictions directly on these cells [21].

1) YOLOv5

YOLOv5 belongs to the YOLO (You Only Look Once) series, renowned for its object detection capabilities in images and videos. It's designed to enhance both performance and speed in identifying objects. This model partitions the image into a network of backbone networks and employs deep neural network techniques to efficiently recognize and categorize objects within the image. A distinctive feature of YOLOv5 is its single-stage architecture, which enables direct object detection from the image without intermediate steps.[22]

Building on the YOLO detection framework, YOLOv5 incorporates several convolutional neural network optimization strategies, such as auto-learning bounding box anchors, mosaic data augmentation, and the cross-stage partial network. The YOLO model set a precedent by being the first object detector to integrate the prediction of bounding boxes with class labels in an end-to-end differentiable network. By leveraging deep learning on extensive labeled datasets, YOLOv5 becomes proficient in identifying a variety of objects, including people, cars, and animals etc., [23].

The architecture of YOLOv5 comprises three main components: the backbone, the neck, and the output. Initially, the input terminal handles data preprocessing tasks like mosaic data augmentation and adaptive image filling. To ensure adaptability to various datasets, YOLOv5 features adaptive anchor frame calculation on the input, automatically setting the initial anchor frame size when the dataset changes. The backbone, such as CSPDarknet, is responsible for extracting fundamental features from the image, while the neck aggregates these features and channels them to the output layers [24].

Known for its high performance and rapid inference speed, YOLOv5 is well-suited for real-time applications.

2) YOLOv8

YOLOv8 marks a significant milestone in the evolution of the YOLO model series, introducing several key advancements. The utilization of anchor-Free-boxes represents a fundamental shift in object detection methodology. By directly predicting object centers, this approach simplifies the model's architecture and expedites the Non-Maximum Suppression (NMS) process during post-processing, thereby enhancing overall efficiency [25]. Additionally, YOLOv8 incorporates the C2f module instead of the previous C3 module. This module combines the outputs of all bottleneck modules, leading to improved model performance. Consequently, the training process is accelerated, and gradient flow is enhanced, resulting in heightened precision and efficiency in object detection tasks [26].

In the architecture of YOLOv8, the backbone network serves as the cornerstone, tasked with extracting features from input images. YOLOv8 adopts CSPDarknet53, a variant of Darknet, as its backbone, introducing a Cross-Stage Partial (CSP) connection to enhance information flow between network stages and improve gradient flow during training. Moving to the neck and head structures, YOLOv8 integrates a Path Aggregation Network (PANet) as the neck, facilitating effective information flow across different spatial resolutions to capture multi-scale features efficiently. The head structure comprises multiple detection heads, each responsible for predicting bounding boxes, class probabilities, and objectness scores at various scales. The true innovation lies in the detection head of YOLOv8, featuring a modified version of the YOLO head that incorporates dynamic anchor assignment and a novel Intersection over Union (IoU) loss function. These enhancements result in more precise bounding box predictions and improved handling of overlapping objects, marking a significant advancement in object detection capabilities [27].

B. Training Methodology

1) Crossing the road Model:

The crossing-the-street model was trained on both YOLOv5 and YOLOv8 architectures with the objective of achieving the highest accuracy between them. The training process utilized a predetermined set of hyperparameters, including a learning rate of 0.001 and a batch size of 16, over the course of 150 epochs. The dataset employed for training comprised 3000 samples. Subsequently, the model's performance was evaluated on a separate test set consisting of 700 samples and validated on an additional set of 700 samples as shown in Table2.

2) Search for Object Model:

The search for object model utilized the YOLOv8 architecture and was trained on the Common Objects in Context (COCO) dataset [19]. This dataset comprises 80 distinct classes and a total of 100,000 images, partitioned into 70,000 training samples, 20,000 validation samples, and 10,000 testing samples. The training of the model involved the specification of hyperparameters, namely a learning rate of 0.001 and a batch size of 16 across 150 epochs, as shown in Table 3.

TABLE 2.THE HYPERPARAMETERS SET DURING CROSSING THE ROAD MODEL MODELTRAINING

Hyperparameter	Value for Yolov8	Value for Yolov5
Input image size	640	640
Epochs	150	100
Batch size	16	16
Optimizer	SGD	SGD
Initial learning rate	0.01	0.01
final learning rate	0.01	0.01
Momentum	0.937	0.937
Weight decay	0.0005	0.0005

TABLE 3.THE HYPERPARAMETERS SET DURING SEARSH FOR OBJECT MODELTRAINING

Hyperparameter	Value for Yolov5
Input image size	640
Epochs	100
Batch size	16
Optimizer	SGD
Initial learning rate	0.01
final learning rate	0.01
Momentum	0.937
Weight decay	0.0005

C. *Training Environment:*

In order to train a model, high computational resources, such as GPUs, are required. We utilized Google Colab, a cloud-based platform that allows the execution of Python code, for training our models. Google Colab provides a free GPU T4 graphics card with a VRAM of 12GB. For some experiments that were taking longer, we opted for the paid version, Colab Pro, which offers more options for powerful GPUs. Colab Pro comes with GPU V100 and GPU A100, which are faster than GPU T4. Upgrading to the Colab Pro version significantly improves the speed of the training process.

D. *Evaluation Metrics*

Evaluation metrics such as precision (P), recall (R), and mAP are commonly used to assess the performance of a model in detecting defective fire extinguishers. These metrics provide a detailed understanding of the model's accuracy and reliability. They are calculated from a confusion matrix that includes four important components:

True Positives (TP): instances correctly classified as positive by the model, belonging to the positive class.
False Positives (FP): instances incorrectly classified as positive by the model, despite belonging to the negative class.

True Negatives (TN): instances correctly classified as negative by the model, belonging to the negative class.

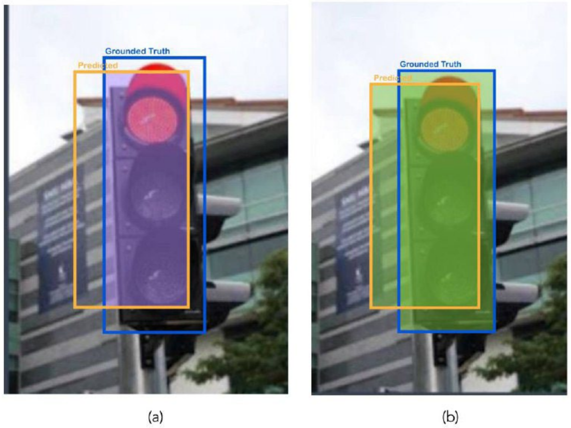
False Negatives (FN): instances incorrectly classified as negative by the model, despite belonging to the positive class.

From these values:

1) $\text{Precision} = \frac{TP}{\text{TP+TF}}$
2) $\text{Recall} = \frac{TP}{\text{TP+TN}}$

Intersection over Union (IoU) is a crucial metric in object detection. It measures the precision of a bounding box by comparing the overlapping areas between the predicted bounding box (Bpr) and the actual bounding box (Bgt) with their combined area. The formula for IoU is given in Equation 3, and to help understand it better, Figure 3 provides a visual representation.

3) $IoU = \frac{B_{pr} \cap B_{gt}}{Bpr \cup Bgt}$



(a) (b)

Fig.3. IoU is the ratio of the intersection area over the union area: (a) Intersection area; (b) Union area.

The mean Average Precision (mAP) is a metric that evaluates the performance of a model. It measures the area under the precision-recall curve for each individual class, represented by Average Precision (AP) [28]. The mAP is then calculated by taking the average of all the individual-class AP values. The IoU (Intersection over Union) threshold of 0.5 is used to indicate the level of overlap required between a predicted bounding box and the ground truth bounding box. This metric is suitable for a broad range of detection applications for both models.

4. RESULTS AND DISCUSSION

A. Mean Average Precision and Model Size

1. Performance Comparison Between YOLOv5s and YOLOv8n Crossing Road Detection Models

Table 4 showcases the performance metrics for YOLOv8n and YOLOv5s in crossing-road function.

TABLE 4.COMPARISON BETWEEN RESULT YOLOV8N AND YOLOV5S MODELS FOR CROSSING-ROAD FUNCTION

Model	Model Size (MB)	P	R	mAP0.5 (%)
YOLOv8n for crossing road	5.9	0.825	0.766	82.4
YOLOv5s for crossing road	13.6	0.753	0.789	85.3

1.1 Mean Average Precision (mAP)

In the crossing-road detection function, the YOLOv5s model outperforms the YOLOv8n model in terms of mean average precision (mAP) at the 0.5 IoU (Intersection over Union) threshold. Specifically, YOLOv5s achieves an impressive mAP of 85.3%, while YOLOv8n trails behind with a lower mAP of 82.4%. This disparity indicates that the YOLOv5s model is more accurate in detecting critical objects related to road crossings, such as crosswalks and traffic lights, compared to the YOLOv8n model.

1.2 Recall (R)

Examining the recall metric, YOLOv5s once again outperforms YOLOv8n. The YOLOv5s model boasts a recall of 0.789, meaning it is able to correctly identify a higher proportion of the relevant crossing-related objects present in the test data. In contrast, YOLOv8n's recall of 0.766 suggests it has a slightly lower ability to detect all the necessary objects for this function.

1.3 Precision (P)

While YOLOv5s excels in mAP and recall, the YOLOv8n model demonstrates superior precision, with a value of 0.825 compared to YOLOv5s's 0.753. This indicates that the YOLOv8n model is more accurate in its detections, producing fewer false positives than the YOLOv5s model.

1.4 Model Size

An important consideration in model selection is the size of the model, which can impact deployment and resource requirements. In this regard, the YOLOv8n model holds a clear advantage, with a compact model size of 5.9 MB. In contrast, the YOLOv5s model is significantly larger, occupying 13.6 MB of storage space. The smaller footprint of the YOLOv8n model may be crucial for applications with limited computational resources or storage constraints.

In summary, the choice between the YOLOv5s and YOLOv8n models for crossing-road detection will depend on the specific priorities and trade-offs between performance metrics and model size requirements. The YOLOv5s model offers higher mAP and recall, while the YOLOv8n model excels in precision and has a more compact model size.

Tables 5 and 6 show the results of YOLOv8n and YOLOv5s, respectively, for each class in the crossing-road and free-walk functions, with the class names.

TABLE 5. DETAILED RESULTS ABOUT YOLOV8N FOR EACH CLASS IN CROSSING-ROAD FUNCTION

Class	P	R	mAP0.5(%)
All	0.825	0.766	82.4
Bicycle	0.872	0.891	91.6
Barrier	0.844	0.772	82.3
crosswalk	0.922	0.962	93.7
Car	0.712	0.725	77.5
green traffic light	0.768	0.631	74.7
red traffic light	0.88	0.812	88.8
yellow traffic light	0.88	0.902	93.7
traffic cones	0.898	0.834	90.2

Pothole	0.764	0.575	65.0
Tree	0.712	0.588	66.4

TABLE 6. DETAILED RESULTS ABOUT YOLOV5S FOR EACH CLASS IN CROSSING-ROAD FUNCTION

Class	P	R	mAP0.5 (%)
All	0.753	0.789	85.3
Bicycle	0.891	0.939	96.6
Barrier	0.796	0.652	81.3
Crosswalk	0.762	0.342	74.1
Car	0.66	0.924	88.1
green traffic light	0.684	0.835	81.9
red traffic light	0.796	0.923	94.1
yellow traffic light	0.791	0.952	95.3
traffic cones	0.777	0.789	85.4
Pothole	0.687	0.804	80.7
Tree	0.685	0.73	0.753

2. Performance Results About YOLOv5s Searching Object Function

Table 7 showcases the performance metrics for the YOLOv5s model designed for searching-object.

TABLE 7. RESULT YOLOV5S MODELS FOR SEARCHING-OBJECT FUNCTION

Model	Model Size (MB)	P	R	mAP0.5 (%)
YOLOv5s for searching object	14.2	0.889	0.923	94.9

2.1 Mean Average Precision (mAP)

The YOLOv5s model designed for crossing road object detection achieves an impressive mean average precision (mAP) of 94.9% at the 0.5 IoU (Intersection over Union) threshold.

2.2 Recall (R)

The YOLOv5s model also demonstrates strong recall performance, with a recall value of 0.923. This means the model is able to correctly identify a very high proportion of the relevant crossing-related objects present in the test data, ensuring that it can detect the majority of important objects in real-world scenarios.

2.3 Precision (P)

In addition to its high mAP and recall, the YOLOv5s model also shows strong precision, with a value of 0.889. This high precision indicates that the model produces relatively few false-positive detections.

2.4 Model Size

The YOLOv5s crossing road detection model has a total size of 14.2 MB.

Table 8. displays detailed training results for the COCO dataset, which consists of 80 classes. It includes precision (P), recall (R), and mAP at 0.5.

TABLE 8. DETAILED RESULTS ABOUT YOLOV5S FOR EACH CLASS IN OBJECT-SEARCHING FUNCTION

Class	P	R	mAP0.5 (%)
all	0.889	0.923	94.9
person	0.972	0.874	95.5
bicycle	0.964	1	99.5
car	0.939	0.672	77.6
motorcycle	0.923	1	99.5
airplane	0.932	1	99.5
bus	0.938	1	99.5
train	0.966	1	99.5
truck	1	0.861	99.5
boat	1	0.876	99.5
traffic light	0.771	0.72	77.5
stop sign	0.857	1	99.5
bench	0.982	1	99.5
bird	0.981	1	99.5
cat	1	0.916	99.5
dog	0.958	1	99.5
horse	0.757	1	99.5
elephant	0.976	0.941	94.9
bear	0.761	1	99.5
zebra	0.905	1	99.5
giraffe	0.98	1	99.5
backpack	0.845	0.912	97.2
umbrella	0.978	0.944	97.8
handbag	1	0.842	89.7
tie	0.943	0.857	85.8
suitcase	0.909	1	99.5
frisbee	0.975	1	99.5
skis	0.797	1	99.5
snowboard	0.837	0.733	91.7
sports ball	0.829	0.667	67
kite	0.972	1	99.5
baseball bat	0.993	1	99.5
Baseball glove	0.638	0.571	65.8
skateboard	0.939	1	99.5
tennis racket	0.784	0.714	79.4
bottle	0.866	0.778	91.8

wine glass	0.756	0.875	90.3
cup	0.92	0.958	97.3
fork	0.928	1	99.5
knife	0.834	0.875	94.9
spoon	0.874	0.947	94.8
bowl	0.858	0.866	89.4
banana	0.803	1	99.5
sandwich	0.752	1	99.5
orange	0.898	1	99.5
broccoli	0.916	1	99.5
carrot	0.826	1	99.0
hot dog	0.841	1	99.5
pizza	1	1	99.5
donut	0.969	1	99.5
cake	0.872	1	99.5
chair	0.943	0.937	98.0
couch	0.913	1	99.5
potted plant	0.973	1	99.5
bed	0.873	1	99.5
dining table	0.992	1	99.5
toilet	0.851	1	99.5
tv	0.84	1	99.5
laptop	0.745	0.667	80.6
mouse	0.878	1	99.5
remote	0.886	0.75	86.3
cell phone	0.945	0.875	87.8
microwave	0.884	1	99.5
oven	0.936	1	99.5
sink	0.774	0.833	88.8
refrigerator	0.919	1	99.5
book	0.951	0.663	88.4
clock	0.961	1	99.5
vase	0.847	1	99.5
scissors	0.212	0.425	49.7
teddy bear	0.987	1	99.5
toothbrush	0.921	1	99.5

B. *Speed*

When it comes to real-time operations, efficiency in terms of time is of utmost importance. Table (9) presents a breakdown of time allocation for two phases, post-processing required for the NMS algorithm, and inference (time taken for passing the image through the neural network).

For the "Crossing Road" models, YOLOv5s demonstrates a total time of 20.4 ms. Conversely, YOLOv8n necessitates a total time of 23.6 ms, these results clearly indicate that YOLOv5s outperforms YOLOv8n in terms of speed, with a lower total time and faster post-processing. Therefore, YOLOv5s showcases superior performance when it comes to the "Crossing Road" function compared to YOLOv8n.

In contrast, concerning the "Search for Object" model, YOLOv5s stands out with a total time of 8.6 ms. Notably, it is the inference time with 8.2ms, while 0.4 ms is dedicated to post-processing.

TABLE 9. DETECTION TIME OF EACH MODEL (USING T4 GPU)

Model	**Inference (ms)**	**Postprocessing (ms)**	**Total Time (ms)**
YOLOv5s (crossing road)	18.4	2	20.4
YOLOv8n (crossing road)	23.2	0.4	23.6
YOLOv5s (Search for object)	8.2	0.4	8.6

C. *Confusion Matrix*

The results of the confusion matrix for each model are presented in Figures (4-6), elucidating the results obtained at a confidence level of 0.25. The diagonal line in the Confusion Matrix represents the instances that have been correctly classified by the model, providing a visual representation of its accuracy. Crosswalk and barrier classes are difficult to detect, which leads to a lower level of accuracy. In addition, there was a small percentage of misclassification in the classification between green, yellow, and red traffic lights.

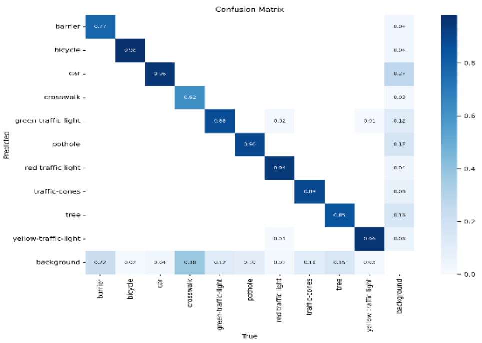


Fig. 4. Confusion matrix for Yolov(crossing road)

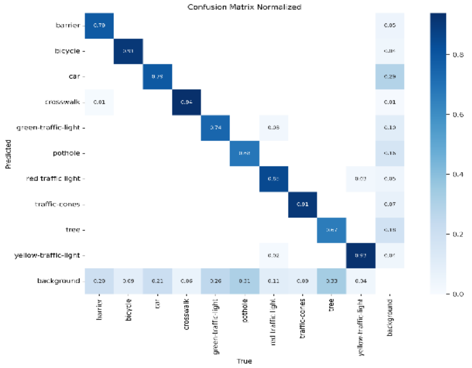


Fig. 5. Confusion matrix for Yolov8 (crossing road)

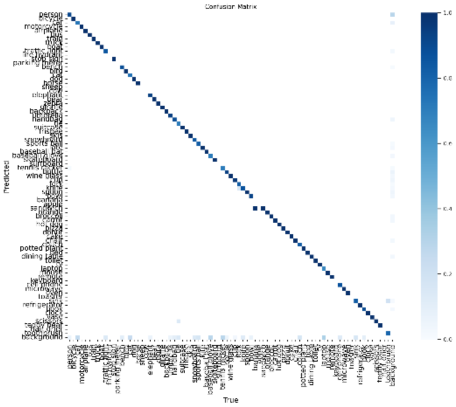


Fig. 6. Confusion matrix for Yolov5 (Search for object

5. CONCLUSION AND FUTURE WORK

The aim of this study is to utilize the YOLO algorithm to assist visually impaired individuals in their daily lives. The study focuses on two main functions: road crossing and object searching. Also, the primary objective was to achieve high accuracy and precision for both models to enable real-time usage on wearable devices and applications.

Both YOLO versions 5 and 8 were trained for the road crossing function using a dataset consisting of 5283 images. The mAP0.5 achieved for YOLOv8n was 82.4%, while YOLOv5s achieved an mAP0.5 of 85.3%. Therefore, YOLOv5s demonstrated better detection performance, and in terms of model size, it is worth noting that YOLOv8n had a smaller model size compared to YOLOv5s, with a size of 5.9 MB for YOLOv8n and 13.6 MB for YOLOv5s. This emphasizes the efficiency and compactness of YOLOv8n, making it more suitable for deployment on resource-constrained devices or systems with limited storage capacity.

The third model, YOLOv5s, was trained for object searching on the COCO dataset, achieving an mAP0.5of 94.9%. It showed excellent results, and its model size was 14.2 MB.

Our future work is focused on enhancing the accuracy and performance of our models, specifically targeting the functionalities related to road crossing and object detection. In terms of the road crossing feature, we aim to expand the training data for both versions 8 and 5 models to encompass a broader range of obstacles that were previously not included in the dataset. While the current dataset primarily focuses on outdoor obstacles like barriers, crosswalks, cars, traffic lights (green, red, yellow), traffic cones, potholes, and trees, we intend to incorporate additional data that includes indoor objects such as doors, stairs, and furniture.

By incorporating a more diverse and comprehensive dataset, we aim to improve the models' ability to accurately recognize and navigate various obstacles in both indoor and outdoor environments. This expansion will contribute to a more robust and reliable system that can assist individuals in safely crossing roads and effectively detecting objects, whether they are indoors or outdoors.

Our ultimate goal is to continuously refine and optimize our models to ensure they provide accurate and reliable assistance, promoting the independence and mobility of visually impaired individuals in navigating their surroundings.

REFERENCES

[1] Coughlan, J., & Manduchi, R. (2009). Functional Assessment of a Camera Phone-Based Wayfinding System Operated by Blind and Visually Impaired Users. International Journal of Artificial Intelligence Tools, 18(3), 379-397. doi:10.1142/S0218213009000196. PMID: 19960101; PMCID: PMC2786081. Retrieved from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2786081/

[2] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2014). ImageNet Classification with Deep Convolutional Neural Networks. In Advances in Neural Information Processing Systems (pp. 1097-1105). https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[3] Girshick, R. (2015). "Fast R-CNN." In Proceedings of the IEEE International Conference on Computer Vision (ICCV), 1440-1448. https://openaccess.thecvf.com/content_iccv_2015/papers/Girshick_Fast_R-CNN_ICCV_2015_paper.pdf

[4] Caldini, A., Fanfani, M., & Colombo, C. (2015). Smartphone-Based Obstacle Detection for the Visually Impaired. In V. Murino & E. Puppo (Eds.), ICIAP 2015, Part I, LNCS 9279, pp. 480-488. Springer International Publishing. DOI: 10.1007/978-3-319-23231-7 43 https://www.researchgate.net/publication/283558670_Smartphone-Based_Obstacle_Detection_for_the_Visually_Impaired

[5] Khenkar, S., Alsulaiman, H., Ismail, S., Fairaq, A., Jarraya, S. K., and Ben-Abdallah, H. (2016) 'ENVISION: Assisted Navigation of Visually Impaired Smartphone Users', Procedia Computer Science, 100, pp. 128-135. doi: 10.1016/j.procs.2016.09.132. https://www.sciencedirect.com/science/article/pii/S1877050916323006

[6] Bai, J., Liu, D., Su, G., & Fu, Z. (2017). A Cloud and Vision-based Navigation System Used for Blind People. In Proceedings of the 2017 International Conference on Artificial Intelligence, Automation and Control Technologies (AIACT '17) (pp. 1-6). New York, NY, USA: Association for Computing Machinery. DOI: 10.1145/3080845.3080867. https://dl.acm.org/doi/abs/10.1145/3080845.3080867

[7] Ghilardi, M. C., Simões, G., Wehrmann, J., Manssour, I. H., & Barros, R.C. (2018). Real-Time Detection of Pedestrian Traffic Lights for Visually-Impaired People. In Proceedings

of the 2018 IEEE International Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 1-10). IEEE. DOI:10.1109/CVPR.2023.00001. Available at: https://ieeexplore.ieee.org/document/8489516

[8] Abdul Muhsin, M., Alkhalid, F. F., & Oleiwi, B. K. (2019). Online Blind Assistive System using Object Recognition. International Research Journal of Innovations in Engineering and Technology (IRJIET), 3(12), 47-51. Retrieved from https://irjiet.com/common_src/article_file/1576904071_76a2f481c3_3_irjiet.pdf

[9] Karmarkar, R. R., & Honmane, V. N. (2021). "Object Detection System for the Blind with Voice Guidance." International Journal of Engineering Applied Sciences and Technology, 6(2), 67-70. https://scholar.google.com/scholar?hl=ar&as_sdt=0%2C5&q=OBJECT+DETECTION+SYSTEM+FOR+THE+BLIND+WITH+VOICEGUIDANCE&btnG=#d=gs_qabs&t=1703018590732&u=%23p%3DY1nCU4aqqekJ

[10] Granquist, C., Sun, S. Y., Montezuma, S. R., Tran, T. M., Gage, R., & Legge, G. E. (2021). Evaluation and Comparison of Artificial Intelligence Vision Aids: Orcam MyEye 1 and Seeing AI. Journal of Visual Impairment & Blindness, 115(4), 277-285. Available at: https://journals.sagepub.com/doi/abs/10.1177/0145482X211027492

[11] Senjam, S. S., Manna, S., & Bascaran, C. (2021). Smartphones-Based Assistive Technology: Accessibility Features and Apps for People with Visual Impairment, and its Usage, Challenges, and Usability Testing. Clinical Optometry, 13, 311-322. DOI: 10.2147/OPTO.S336361. Available at: https://www.tandfonline.com/doi/full/10.2147/OPTO.S336361

[12] Salunkhe, A., Raut, M., Santra, S., & Bhagwat, S. (2021). Android-based object recognition application for visually impaired. *ITM Web of Conferences*, 40, 03001. [Online]. Available: https://doi.org/10.1051/itmconf/20214003001

[13] See, A. R., Sasing, B. G., & Advincula, W. D. (2022). A Smartphone-Based Mobility Assistant Using Depth Imaging for Visually Impaired and Blind. *Applied Sciences*, 12(6), 2802. [Online]. Available: https://doi.org/10.3390/app12062802.

[14] Patil, R., Modi, R., Parandekar, A., & Deone, J. B. (2022). Designing mobile application for Visually Impaired and Blind Persons. SSRN. [Online] Available at: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4108763.

[15] Birambole, A., Bhagat, P., Mhatre, B., & Abhyankar, A. (2022). "Blind Person Assistant: Object Detection." International Journal for Research in Applied Science & Engineering Technology (IJRASET), 10(3). Retrieved from blind-person-assistant-object-detection (ijraset.com)

[16] Busaeed, S., Mehmood, R., & Katib, I. (2022). Requirements, Challenges, and Use of Digital Devices and Apps for Blind and Visually Impaired. NOT PEER-REVIEWED. Preprints [Online]. Available at: https://www.preprints.org/manuscript/202207.0068/v1

[17] Kuriakose, B., Shrestha, R., & Sandnes, F. E. (2023). DeepNAVI: A deep learning-based smartphone navigation assistant for people with visual impairments. Expert Systems With Applications, 212, 118720. DOI: 10.1016/j.eswa.2022.118720. Available at: https://www.sciencedirect.com/science/article/pii/S0957417422017432

[18] Sarmah, A. J., Bhagawati, K., Duwarah, K., Purkayastha, S. D., Boro, A., & Muchahary, D. (2023). "Object detection and conversion of text to speech for visually impaired." ADBU-Journal of Engineering Technology, 12(2), 0120204049 https://scholar.google.com/scholar?hl=ar&as_sdt=0%2C5&q=Object+detection+and+conversion+of+text+to+speech+for+visually+impaired&btnG=#d=gs_qabs&t=1703018313517&u=%23p%3DxYgBPUmyrzUJ

[19] COCO Consortium. COCO - Common Objects in Context. Retrieved from https://cocodataset.org/#home

[20] Roboflow. (n.d.). Roboflow. Retrieved from https://roboflow.com/

[21] Jiang, P., Ergu, D., Liu, F., Cai, Y., & Ma, B. (2022). A Review of Yolo Algorithm Developments. Procedia Computer Science, 199, 1066-1073. doi:10.1016/j.procs.2022.01.146

[22] Benjumea, A., Teeti, I., & Cuzzolin, F. (2022). YOLO-Z: Improving small object detection in YOLOv5 for autonomous vehicles.

[23] Horvat, M. and Gledec, G. (2022) 'A comparative study of YOLOv5 models performance for image localization and classification', Proceedings of the Central European Conference on Information and Intelligent Systems, pp. 349-357

[24] Solawetz, J. (2020). Yolov5 new versionimprovements and evaluation. Roboflow. Seach date. Retrieved fromhttps://blog.roboflow.com/yolov5-improvementsand-evaluation/

[25] Roboflow. (n.d.). What's New in YOLOv8? Roboflow Blog. Retrieved from https://blog.roboflow.com/whats-new-in-yolov8/#yolov8-architecture-a-deep-dive

[26] Viso AI. (n.d.). YOLOv8 Guide. Retrieved from https://viso.ai/deep-learning/yolov8-guide/

[27] YOLOv8 Architecture Overview. (n.d.). YOLOv8. Retrieved from https://yolov8.org/yolov8-architecture/#2_YOLOv8_Architecture_Overview

[28] R. Szeliski, Computer Vision: Algorithms and Applications. Cham, Switzerland: Springer International Publishing, 2022.