Available online at www.ijournalse.org

Emerging Science Journal

(ISSN: 2610-9182)

Vol. 7, No. 5, October, 2023

The Eye: A Light Weight Mobile Application for Visually Challenged People Using Improved YOLOv5l Algorithm

Kalaiarasi Sonai Muthu Anbananthen [1*], Sridevi Subbiah [2], Subiksha Gayathri Baskar [2], Ratchana Selvaraj [2], Jayakumar Krishnan [1], Subarmaniam Kannan [1], Deisy Chelliah [2]

[1] Faculty of Information Science and Technology, Multimedia University, Selangor, Malaysia.

[2] Thiagarajar College of Engineering, Madurai, Tamilnadu, India.

Abstract

The eye is an essential sensory organ that allows us to perceive our surroundings at a glance. Losing this sense can result in numerous challenges in daily life. However, society is designed for the majority, which can create even more difficulties for visually impaired individuals. Therefore, empowering them and promoting self-reliance are crucial. To address this need, we propose a new Android application called “The Eye” that utilizes Machine Learning (ML)-based object detection techniques to recognize objects in real-time using a smartphone camera or a camera attached to a stick. The article proposed an improved YOLOv5l algorithm to improve object detection in visual applications. YOLOv5l has a larger model size and captures more complex features and details, leading to enhanced object detection accuracy compared to smaller variants like YOLOv5s and YOLOv5m. The primary enhancement in the improved YOLOv5l algorithm is integrating L1 and L2 regularization techniques. These techniques prevent overfitting and improve generalization by adding a regularization term to the loss function during training. Our approach combines image processing and text-to-speech conversion modules to produce reliable results. The Android text-to-speech module is then used to convert the object recognition results into an audio output. According to the experimental results, the improved YOLOv5l has higher detection accuracy than the original YOLOv5 and can detect small, multiple, and overlapped targets with higher accuracy. This study contributes to the advancement of technology to help visually impaired individuals become more self-sufficient and confident.

Keywords:

Text to Speech;

Visually Challenged;

YOLO;

Machine Learning.

Article History:

Received:	01	June	2023
Revised:	27	August	2023
Accepted:	09	September	2023
Published:	01	October	2023

1- Introduction

A sizeable fraction of the world’s population suffers from vision impairment, with 36 million completely blind and 216 million having moderate to severe visual impairments [1]. They face challenges in perceiving their surroundings. These people frequently rely on other senses, such as touch or aural signals, to navigate their environment. Recognizing items can be challenging and, occasionally, even harmful without the capacity to touch them. To address this issue, this article proposes an Android application [2] designed specifically to support visually challenged individuals. By leveraging machine learning-based object detection techniques, the mobile application [3] can recognize a wide range of objects in real-time, providing invaluable assistance to individuals with visual impairments. This technology enables users to quickly and accurately identify objects in their environment without touching them. Our study aims to create a reliable and efficient solution that empowers visually challenged individuals, promotes independence, and enhances their overall quality of life.

[*] **CONTACT**: kalaiarasi@mmu.edu.my

DOI: http://dx.doi.org/10.28991/ESJ-2023-07-05-011

In modern society, the smartphone has transformed from a simple communication device to a powerful computing tool that simplifies daily activities. A survey conducted by Ericsson on smartphone usage reveals a significant increase in the number of individuals who use smartphones. Figure 1 illustrates the projected increase in smartphone users, with an estimated 7,516 million people expected to use smartphones by the end of 2026. This data highlights the growing significance of smartphones in our lives and underscores the importance of leveraging their capabilities to develop innovative solutions that can address societal needs.

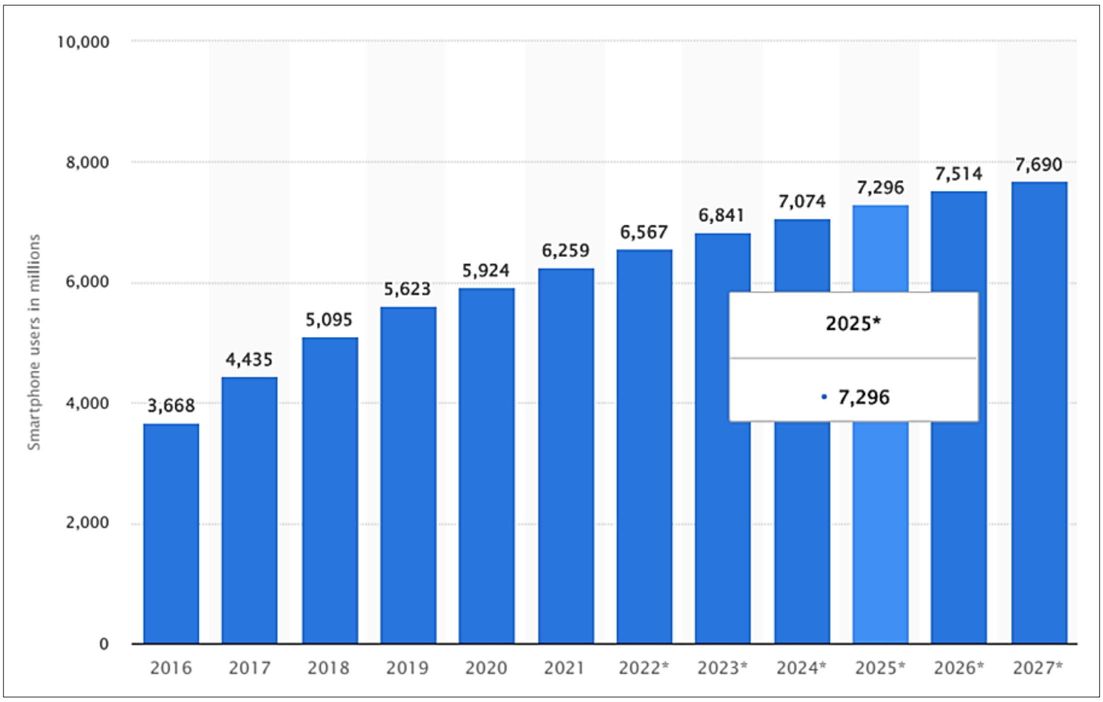


Figure 1. Smartphone subscriptions worldwide (*https://www.statista.com* [4])

A survey conducted on 259 individuals who are blind revealed that a significant number of participants, approximately 82.2%, had been using their mobile devices for more than two years. Interestingly, the experience of mobile device usage was consistent across age groups and visual function categories. Furthermore, the survey indicated that 90.3% of respondents used paid and free applications, while 83.1% perceived mobile applications as user-friendly, and 80.7% found them accessible. Additionally, over 90% of participants reported using applications specifically designed for individuals with visual impairments. These survey results [5] show that visually challenged individuals prefer tools that are specifically designed for their needs. This highlights the importance of developing solutions tailored to their requirements.

This article proposes the development of a mobile-based application named “The Eye” to address the needs of visually challenged individuals and provide them with a sense of independence. The application aims to function as the eyes of visually challenged individuals by enabling them to identify objects in real-time using their Android mobile camera. Previous research by Huu et al. [6] and Wu et al. [7] have explored the application of YOLOv4 and YOLOv5 object detection algorithms in smart education and complex scene analysis, respectively. Mobile applications designed for visually challenged people enhance their accessibility, object recognition [8, 9], social interaction and communication, education and learning, independent living etc. The YOLO (You Only Look Once) object detection algorithm [3, 10, 11] has evolved, including YOLOv4 [12] and YOLOv5. YOLOv4 improved accuracy and introduced architectural enhancements, while YOLOv5 simplified the architecture and prioritized real-time inference. Both versions have contributed to the advancement of object detection algorithms and have been widely adopted for various computer vision applications.

However, the existing literature indicates that YOLOv3, YOLOv4, and YOLOv5 algorithms may struggle to effectively capture complex features and details. To overcome these drawbacks, the article adopts an advanced and customized version of YOLOv5l (You Only Look Once Version 5 with a larger variant) as the base architecture for object detection. A Tensor-Flow Lite model is built upon this customized YOLOv5l architecture to enhance its performance further. YOLOv5l, which is a larger variant, has a larger model size and can capture complex features and details with improved object detection accuracy compared to smaller variants like YOLOv5s and YOLOv5m. Notably, YOLOv5l requires more computational resources for training and inference, making it suited for applications where higher accuracy is prioritized over faster inference times and when ample computing resources are available. The choice between YOLOv5 and YOLOv5l is based on application-specific requirements, available resources, and the trade-off between accuracy and speed. Although YOLO5l doesn't explicitly use Spatial Pyramid Pooling (SPP), its larger receptive field and increased feature representation of YOLOv5l contribute to its ability to manage objects of varying scales and capture fine-grained details.

The proposed approach uses an improved object detection model, YOLOv5l, to detect objects, which is then processed to obtain a count for each class. The output count is then converted to a voice message that is delivered to the user. The proposed solution allows visually challenged individuals to perceive their surroundings and navigate their environment more easily.

The primary objectives of this research are

- To perform image classification that captures images in real-time using various image classification methodologies such as Recurrent neural network (RNN), YOLOv4, and improved YOLOv5l.
- To detect objects in real-time through the Android mobile's camera and convert them to voice messages.
- To determine the best methodology for designing an application that assists visually challenged people.

This paper is organized as follows: Section 2 describes related research works and existing mobile applications, section 3 outlines dataset details, and Section 4 contains the proposed design methodology. Results and discussions are presented in Section 5, followed by the conclusion and the scope for future work in Section 6.

2- Literature Review

Fu [2] proposed a mobile assistant application for visually challenged people, including face detection, gender classification, and audio representation of images. The author deals with an app that completely implements face detection and gender classification in offline mode. The app uses a Convolutional Neural Network (CNN) for the gender classification process. Rather than using the existing Fisher faces or Egen faces, the app uses deep learning to carry out face detection and gender classification process. These methods have low accuracy when people don't face the camera. However, since this can be the case in the real world, the app uses deep learning to overcome this drawback. ImageNet mobile model has been used to coach their model.

Lara and Miguel presented a real-time human activity recognition approach on an Android mobile platform [5]. The proposed method integrates wearable sensors with a mobile application to monitor human actions. The authors utilized a library called Mobile Evaluation of Classification Algorithms (MECLA) and a mobile app to enable real-time human action recognition within a Body Area Network. Furthermore, the authors employed MECLA to assess the performance of various decision tree classification algorithms on mobile devices.

Mao et al. [3] discussed Real-Time Object Detectors for Embedded Applications. In the paper, they speak about a project for real-time object detection in embedded systems. They built their model with Darknet-53 as the base. Their main work focuses on a feature extraction backbone network that incorporates a parameter size of 16% of dark-net-53. They must diminish the network's parameter size by utilizing separable depth and point-wise group convolutions to enhance the accuracy. They added a Multi-Scale Feature Pyramid Network that works supported by a simple U-shaped structure. Anbananthen et al. [13] and Subbiah et al. [14] used feature selection methods and hybrid machine learning algorithms for crop yield prediction. Sridevi et al. [15] used pattern-matching algorithms for predicting time series data. Pattern matching algorithm includes K-Means and K-Medoids algorithms.

Hu et al. [10] proposed a video-streaming vehicle detection method based on YOLOv4. They used the YOLOv4 methodology to detect vehicles and enhanced the methodology's speed without sacrificing its accuracy. The real-time object detection method is based on improved YOLOv4-tiny by Jiang et al. [11]. The author used an improved YOLOv4 methodology to improve the speed of real-time object detection. The YOLOv4-tiny method uses the Cross Stage Partial (CSP) Block module as its residual module. While it helps improve accuracy, it also increases the network complexity, thereby reducing the speed of object detection. To boost the speed of object detection with a small impact on its accuracy, they proposed an improved YOLOv4 methodology. The proposed methodology uses the ResBlock-D module rather than two CSP Block engines. To balance the object detection time and accuracy, they designed an auxiliary network block using two 3×3 convolutions networks, channel attention, spatial attention, and concatenate operation to extract global features. They merged their designed auxiliary network into the backbone network to create a different backbone network. Wu et al. [16] proposed a channel pruning-based YOLOv4 deep learning methodology for the real-time and accurate recognition of apple flowers in natural environments [16]. They began their process by building the YOLO v4 model under the Cross Stage Partial (CSP)Darknet53 framework. They used the channel pruning methodology to prune the model and enhance its efficiency. They fine-tuned their model with a set of 2230 manually labeled apple flower images and have achieved a quick and accurate detection model for detecting apple flowers.

Ramiz and Derya (2021) tracked On-Shelf (OSA) Availability using the YOLOv4 deep learning architecture. This paper also offers the first explainable artificial intelligence (XAI) OSA demonstration. It introduces a brand-new software programme with the features and benefits of semi-supervised learning and on-shelf availability (SOSA) and XAI. "YOLOv4 has been widely used in many different fields such as health [17, 18], marketing [19], marine [20], agriculture [7, 21], and education [6]. Linglin et al. [12] developed a pruning approach for the target identification model using YOLOv4. The authors used L1 regularisation of the channel scale factor in this research, which resulted in sparse channels in the convolutional layer. They proved that the pruned YOLOv4 model could better detect targets on their chosen dataset than the regular model. Compared with the existing YOLOv4, there was a little loss of accuracy. The pruned model's volume, parameter amount, and reasoning time were greatly reduced, compensating for the accuracy loss. Pruning made the model more compressed and performance better [22].

An Effective and Efficient Object Detector for Autonomous Driving using YOLOv4 was proposed by Cai et al. [23]. They suggested a one-stage object detection framework based on the YOLOv4 model for improving detection accuracy while supporting real-time operation. The proposed framework's backbone network is CSP Darknet53 DCN (deformable convolution). However, the network's final output layer is replaced with deformable convolution to improve detection accuracy. They developed a feature fusion module called Pixel Aggregation Network (PAN) to improve the detection accuracy of small objects.

Ponnada et al. [24] introduce a model for visually challenged people to identify the bus number, destination, the bus doors and the path for reaching the bus. Bagyam et al. [25] proposed CNN and RNN for object classification, which is then converted into an audio message. Kumar [26] developed an interface that randomly plays a song using instrumental music. Their system used visual and sound Completed Automated Public Turing tests to differentiate Computers and Humans (CAPTCHA) to achieve the goal.

Ranjan & Navamani [27] built an android-based application called Blind Learning APP (BL-APP) to help visually challenged persons in their learning processes. This programme (App) has a user-friendly interface and a variety of built-in learning materials. Arvind & Hole [28] proposed a CNN and RNN for object classification. The author used an Android application for people with visual impairments to demonstrate the usefulness and practicality of the Neural Network in the real world.

Bhatia et al. [29] used CNN with Rectified Linear Units (ReLU) to recognize Arabic speech in real-time and convert it to Arabic text. Participants with vision and hearing impairments who were skilled at reading Braille could decipher the Braille lettering that was triggered on the fingers. Baskar et al. [30] developed a portable embedded device with face recognition capabilities designed to provide audio feedback to visually impaired people to help them recognize faces. This device utilizes the LAB color space and Contrast Limited Adaptive Histogram Equalization (CLAHE) with gamma enhancement for improved accuracy. Al-Allaf et al. [31] used Particle Photon with many sensors, a Global Positioning System (GPS) unit, and auxiliary components for alarm. This system enables people to detect objects and instantly contact their parents or friends in case of emergency with the Blynk application by pressing the SOS button (Save our Selves). The literature survey of existing mobile applications is shown in Table 1.

Table 1. Literature survey of existing applications

App name	Functionality
KNFB reader or OneStep Reader by National Federation of the Blind and Sensotec NV	It is an app that converts text to speech and speech to text to Braille. It helps in reading receipts, documents, product information, etc. It just focuses on text detection and is not free of cost. For the app to function as intended, it requires a $100 purchase and additional in-app purchases.
Tap Tap See	This application utilizes the device camera to detect the user's desired image. It requires the user to tap when the image when they want in the camera and uses the Cloud Sight Image Recognition API to identify and convert it into speech. But it can only recognize objects in focus and within the camera's scope. The Lighting conditions also play a major role in the quality of the object identification.
Seeing Artificial Intelligence (AI) by Microsoft	The app has multiple channels dedicated to a feature like text reading, object recognition, currency recognition, and more. The app utilises Microsoft's Artificial Intelligence (AI) and cloud services. It is free of cost but is only available for IoS users with iPhone SE or the iPhone 6 and above. It also requires the internet to be available.
Be my eyes	Be my eyes is an application that connects volunteers with visually impaired people. The app requires visually impaired people to request assistance, and a volunteer will connect through a live video call to help them out. This app highly depends on the availability of volunteers and the user's network connection to ensure proper video clarity.
Aipoly vision	It is an object and color recognizer app that utilizes Artificial Intelligence (Artificial Intelligence (AI)) to recognize the object on camera and announce it to the user. The app has a few daily and necessary object classes for free users. It is available in its complete form for the IoS users alone, and forgetting its full functionality, the app includes in-app purchases.
Lookout by Google	Lookout is an app for Android devices that utilizes computer vision to identify the details about objects, documents, etc. It utilizes the camera and sensors available in a smartphone to provide the above functionalities. It has two modes, food label, and scan mode, that can be switched according to daily needs. It utilizes TensorflowLite and Google on-device machine learning solutions.

A sparse scaling factor has also been accustomed to improving the present channel pruning methodology. A novel mobile application to assist the visually impaired was proposed by Khan Shishir et al. [32]., They proposed an analogous approach to develop an app that uses TensorFlow API to detect objects in real-time and the Text to Speech (TTS) module of Google to output the result as voice. They began by segmenting the object from its background using the TensorFlow machine learning API (Application Programming Interface) and then formulated the recognition problem as an instance retrieval task. Finally, a text-to-speech module informs the user of the object's identity.

3- Dataset

The YOLOv5 model uses the Common Objects in Context (COCO) dataset [33], the most commonly used dataset for object detection segmentation and captioning. On a Pascal Titan, X YOLO can process images at 30 fps. It is a large-scale dataset containing about 330K images; 200K are labeled. The dataset includes 80 categories of objects like animals, flowers, buses, persons, etc. The COCO dataset contains 80,000 training images and 40,000 validation images. An advantage of the COCO dataset in object detection is that it provides bounding boxes and per-instance segmentation masks for these 80 object classes. It also contains more than 200,000 images and 250,000 person in-stances labeled with key points that can be used to train a detection model effectively.

Another advantage of COCO is its natural image collection that reflects daily life scenes. There will be multiple objects in a single frame in everyday life scenes, which should be segmented and labeled properly. With the help of the COCO dataset, the model can be thus trained on classifying multiple images from a single scene, suiting it for daily needs. The COCO dataset has 80 classes and their labels. Some of the sample class labels are shown in Table 2. Figure 2 depicts sample images from the COCO dataset.

Table 2. Coco dataset label sample

ID	OBJECT (PAPER)	OBJECT (2014 REL.)	OBJECT (2017 REL.)	SUPER CATEGORY
1	person	person	person	person
2	bicycle	bicycle	bicycle	vehicle
3	car	car	car	vehicle
4	motorcycle	motorcycle	motorcycle	vehicle
5	airplane	airplane	airplane	vehicle
6	bus	bus	bus	vehicle
7	train	train	train	vehicle
8	truck	truck	truck	vehicle
9	boat	boat	boat	vehicle
10	traffic light	traffic light	traffic light	outdoor

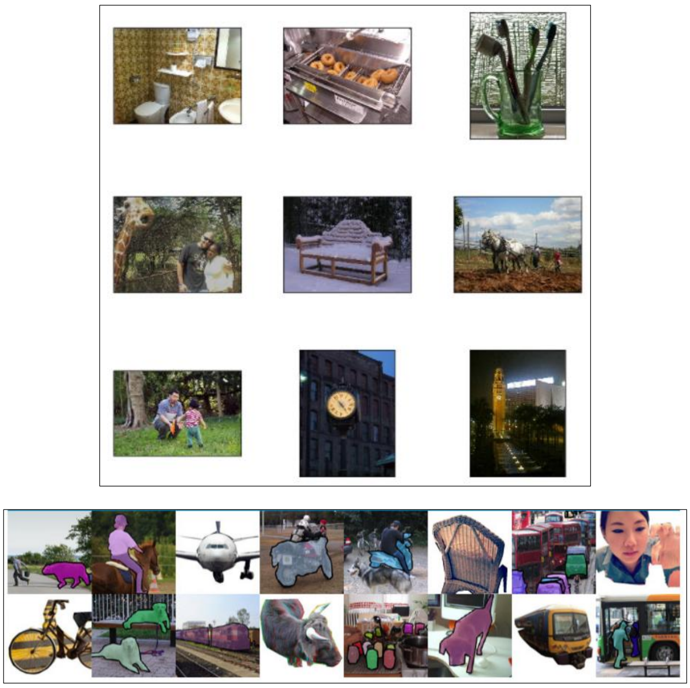


Figure 2. Sample images from the COCO dataset

4- Research Methodology

Though existing mobile applications share similarities with our proposed application, they come with various features that need payment for utilization. What sets our proposed application apart is utilizing the YOLOv5l object detection model for object detection. This methodology is considered the YOLO model's fastest and most accurate version. The research employs various image processing techniques, such as cropping and adjusting the exposure and brightness, to give the model a more understandable image. Our app primarily focuses on real-time object detection. It provides the users with a voice output that tells them about the detected objects and their count, making it easier for them to understand their environment. The app can be opened with the help of Google Assistant, and nothing is needed from the user side, making it comfortable to use by the visually impaired. It works on a real-time basis, thereby saving memory and time of response. This way, the user need not worry about clicking a proper image or selecting the appropriate function they want the app to perform. It simply requires the user to open the app to perform object detection and render the result.

4-1-App Workflow

Our Android application runs on a TFLite model trained by pre-trained weights and customized by modifying the configurations based on our needs and performance criteria. This model works on real-time image inputs labeled with object names bounded by boxes. The Workflow depicted in Figure 3 gives a basic idea of how the image is processed and identified. The user can do real-time object detection with the help of their back camera. Initially, the input image is pre-processed using image processing techniques, including cropping the image based on a pre-defined or user-defined size. Cropping provides a better focus on the objects rather than focusing on unwanted things. The prediction function will be called upon extracting, using a TFLite model based on the YOLOv5l model trained with the COCO dataset and configured. We have used pre-trained weights of the COCO-trained TensorFlow model and generated TensorFlow lite, which adheres to YOLOv5l architecture. This methodology automatically finds the object bounded by boxes in the image and labels it with the respective class name. Once the detection is done, the result list containing the labels is iterated over to count the occurrence of each class. These count results are then sent to Android's Text to Speech class that renders the voice output.

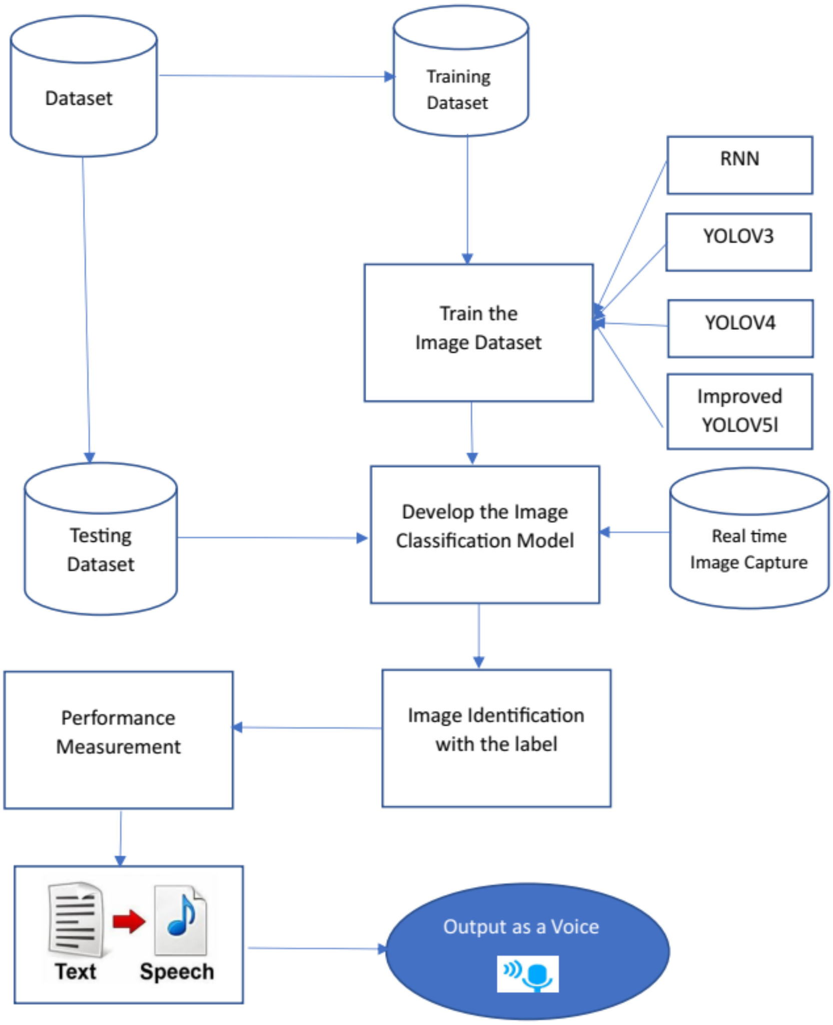


Figure 3. **Block diagram of "The Eye"**

4-2- Data Collection

As a first step in developing the TFLite model, we explored numerous datasets for object detection. This includes the COCO dataset, Open images, and CIFAR-10 dataset. The COCO dataset is large-scale object detection, segmentation, and captioning dataset. Open image is a dataset of around 9M images annotated with image-level labels and object-bounding boxes. The CIFAR-10 dataset contains 60,000 32x32 colour images divided into 10 classes, each with 6000 images. There are 50,000 training and 10,000 test images, to name a few. After evaluating these datasets, it is decided to use the COCO dataset, which has more classes and supports real-time object detection by having images that can be found in real life with more than one object in a frame.

4-3- Model Architecture

The proposed model was built using YOLOv5. Here, the object detector is based on the YOLOv5l object detection model. YOLOv5l is the larger variant of the YOLOv5 family, focusing on accuracy at the expense of inference speed and memory consumption. It has 253 layers and a width multiplier of 1.0, resulting in a model size of about 90 MB. It is considered popular architecture because of its speed, with the base model having a speed of 45 frames per second and the fast model having a speed of 155 frames per second, which is 1000x faster than Recurrent Convolutional Neural Networks (RCNN). The architecture (shown in Figure 4) comprises a single neural network that can learn the general representation of objects.

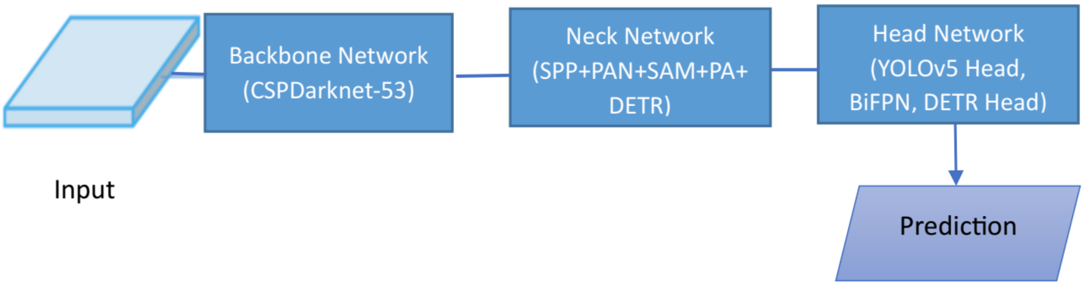


Figure 4. Architectural view of YOLOv5

The YOLOv5l architecture comprises three primary components: the Backbone Network, the Neck Network, and the Head Network [8, 9]. The CSPDarknet-53 architecture forms the Backbone Network, a modified version of the Darknet architecture that includes cross-stage partial connections to enhance the flow of information between different layers. The Neck Network combines various modules, such as Spatial Pyramid Pooling (SPP), Path Aggregation Network (PAN), Spatial Attention Module (SAM), Positional Attention (PA), and Detection Transformer (DETR), to extract rich features and improve the model's performance. The Head Network generates the final predictions, combining the YOLOv5 Head, Bi-directional Feature Pyramid Network (BiFPN), and DETR Head to predict the class labels and bounding boxes for objects in the input image. Finally, the Prediction component takes the Head Network's output and produces the final predictions.

If the model has a single-stage detector, the head makes dense predictions. The final prediction is a vector comprising the predicted bounding box coordinates, including the centre, height, and breadth, the confidence score for the prediction, and the probability classes.

The functions of YOLOV5 are as follows:

1. Load the pre-trained YOLOv5 model
2. Capture the video or image frame
3. Resize the frame to the desired input size of the YOLOv5 model
4. Convert the frame to tensor format and normalize the pixel values
5. Pass the tensor through the YOLOv5 model to get the predicted bounding boxes, class probabilities, and confidence scores for each object detected in the frame.
6. Apply non-maximum suppression (NMS) to remove redundant bounding boxes and keep only the most confident ones.
7. Draw the final bounding boxes and class labels on the original frame
8. Display the frame with the object detection.

YOLOv4 and YOLOv5 are state-of-the-art object detection models developed by the same research group. Here are some key differences between the two models: Model Architecture: YOLOv4 uses a modified Darknet architecture with CSP (cross-stage partial) connections, whereas YOLOv5 uses a new CSPNet based on CSP blocks. CSPNet has fewer parameters and is faster than YOLOv4.

- Backbone Network: YOLOv4 uses the CSPDarknet53 backbone network, which has 53 convolutional layers, whereas YOLOv5 uses the CSPResNeXt50 backbone network, which has 50 layers.
- Training Data: YOLOv4 was trained on a combination of COCO and various other datasets, whereas YOLOv5 was trained on the same COCO dataset but with additional synthetic data generated using a technique called CutMix.
- Inference Speed: YOLOv5 is faster than YOLOv4 due to its optimized CSPNet architecture and anchor-free detection.
- Accuracy: Both YOLOv4 and YOLOv5 achieve state-of-the-art accuracy on the COCO dataset, with YOLOv5 achieving slightly better results in terms of mAP (mean average precision).

Overall, YOLOv5 is faster and more accurate than YOLOv4, but it is important to note that the choice of model depends on the specific application requirements and the available computing resources. The larger variant of YOLOv5 is used in the proposed work.

YOLOv5l includes several other key features, including:

- A feature pyramid network (FPN) combines low-level and high-level features to improve detection accuracy at different scales.
- A PAN (path aggregation network) head that further improves feature fusion and aggregation.
- A set of anchor boxes that are used to define the aspect ratios and sizes of the objects to be detected.
- A classification and regression sub-network that uses a series of fully connected layers to predict the class labels and bounding boxes for each object in the input image.
- Overall, the YOLOv5l architecture is designed to be highly scalable and efficient, with relatively few parameters compared to other state-of-the-art object detection models, while still achieving high accuracy and precision.

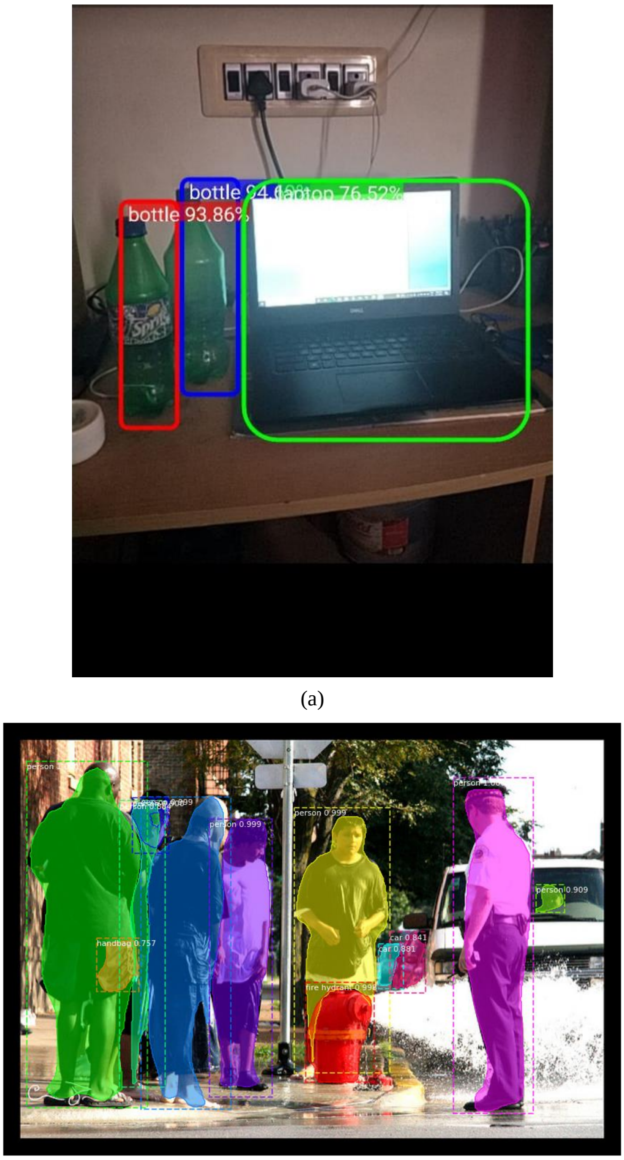
The proposed work introduces an improved version of the YOLOv5l algorithm that incorporates L1 and L2 regularization. L1 and L2 regularisation is often used in deep learning models to reduce overfitting and enhance the model's generalizability. During the training process, these techniques involve the addition of a regularisation term to the loss function.

L1 regularization, also known as Lasso regularization, adds the sum of the absolute values of the model's weights to the loss function. Doing so encourages the model to attribute less importance to less significant features, effectively reducing their significance to zero. This approach to regularisation promotes model simplicity and prevents excessive reliance on specific features, thereby facilitating improved generalization.

In contrast, L2 regularisation, also known as Ridge regularisation, integrates the sum of the weights' squared values into the loss function. It penalizes greater weights, promoting a more balanced distribution of feature importance. L2 regularisation prevents the model from overemphasizing individual features, resulting in more uniform and robust solutions. Higher lambda values enhance the regularisation effect, whereas lower values permit the model to rely more heavily on the original loss function.

5- Results and Discussion

YOLOv5 was trained on the COCO dataset, and it can detect 80 classes. We have set the subdivisions to 32 to split our batches into 32 mini-batches to increase the speed of the result. The proposed work used pre-trained weights for better accuracy. Furthermore, we have set the width and height to 416, which has helped the model focus on the image clearly, increasing the accuracy of the result and reducing detection speed. Then we converted the TF (TensorFlow) model into the TFLite model in order to use them in an Android application. This model was then included in the detector activity that gets called when the camera is on. The input image was processed before being fed into the trained model. The model produces a list of labels for all items in the scene. This list was then used to count the number of times each class label appeared in the result. The label and its occurrence count were stored as key-value pairs in HashMap and passed to the Android Text to Speech class as a parameter. This class takes the text as input and maps it to sound. This sound is then given as the user's output, completing the entire process from real-time object detection to voice output. Figures 5-a and 5-b depicts detected objects using the proposed methodology.



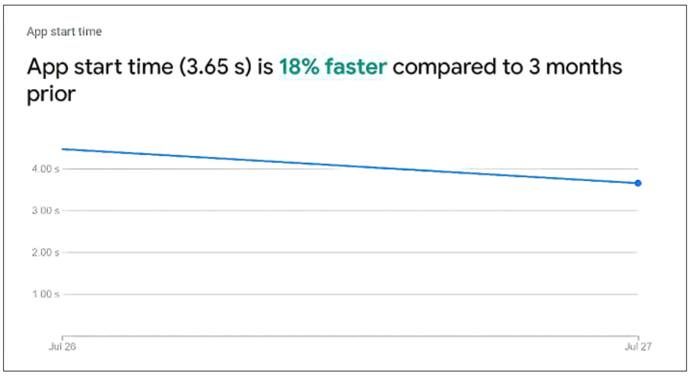
(a)

(b)

Figure 5. a) Objects detection using our app, b) Person and objects detection using YOLOv5l

The application was installed and tested on Redmi 5A, Realme, Redmi 6 Pro, and Samsung M31 mobile phone models. During testing, the application on all devices demonstrated accurate and rapid real-time object detection via the rear camera. Post-processing techniques, such as non-maximum suppression (NMS), enhanced the final detection results. NMS successfully eliminated duplicate or overlapping bounding frames by selecting the most confident detections and suppressing redundant detections. The voice output, which indicated the number of labels for each class, was clearly conveyed. Figures 5-a and 5-b illustrate the application's ability to detect objects and identify individuals.

Firebase was integrated into our application to analyze its performance comprehensively. Firebase is a powerful and comprehensive platform provided by Google that offers various services and tools to facilitate the development of mobile applications. It provides a range of features that can be utilized in mobile app development, including authentication, cloud firestore, analytics, cloud functions, etc. Firebase Cloud Storage provides an easy-to-use and scalable solution for storing and serving user-generated content such as images, videos, or other files. It offers secure and reliable cloud storage with built-in access controls, making it convenient for mobile apps that require file storage and sharing capabilities. Firebase Analytics helps to understand user behavior and measure the mobile app's performance. Firebase Performance Monitoring helps monitor and analyze the mobile app's performance. It provides insights into app startup times, network requests, method traces, and resource consumption, allowing you to identify performance bottlenecks and optimize app performance. Firebase offers SDKs for various platforms, including iOS and android, making integrating these services into mobile app development workflow easy. It simplifies backend infrastructure management, reduces development time, and provides a robust and scalable backend solution for mobile apps. By assessing the app's page rendering capabilities, we reduced the app's startup time, resulting in an 18% improvement, as demonstrated in Figure 6.



App start time

App start time (3.65 s) is 18% faster compared to 3 months prior

Figure 6. Performance measurement based on metrics specified

5-1- Result

In object detection, the mean Average Precision (mAP) is a commonly used evaluation metric to measure the precision and performance of various models, including YOLO. The mAP metric evaluates the model's ability to reliably classify and localize objects by combining precision and recall. It is a useful instrument for comparing the performance of various object detection models, including different variants of YOLO and determining their suitability for particular datasets and applications. Higher mAP values indicate superior object detection accuracy and localization capabilities. In addition, Frames Per Second (FPS) is a frequently employed metric for measuring the speed or efficacy of object detection models such as YOLO. FPS measures the number of frames or images the model can process in one second. YOLOv5 is an improved model in terms of accuracy measured in mean Average Precision (mAP) and speed measured in frames per second (FPS). Figure 7 shows the performance of YOLOv5 in Real-time for a COCO dataset-trained model. The RNN model was trained with the same dataset as the input, resulting in an accuracy of 70.4% mAP. The training and Validation loss of RNN is shown in Figure 8.

YOLOv3 was trained and tested, which resulted in an accuracy of 78.31%. YOLOv5 was trained and tested, which resulted in an accuracy of 78.4%. Then various configuration modifications were tried on the YOLOv5l model; one of them was setting the subdivisions to 32, with the width and height set at 416 and 512, respectively. We also tried changing the activation functions of various convolution layers, then finalized our configurations as mentioned in the methodology and obtained an accuracy of 88.4%. The training and validation loss of RNN is shown in Figure 8. The efficiency of the work is compared with YOLOv4 [20], improved YOLOv4 [20], and Hough Transform algorithm [12]. YOLOv5l has a larger model size, which allows it to capture more complex features and details. It offers improved object detection accuracy compared to smaller variants like YOLOv5s and YOLOv5m. The improved YOLOv5l works based on L1 and L2 regularization. In YOLOv5l, L1 and L2 regularization are regularization techniques used to prevent overfitting and improve generalization. They work by adding a regularization term to the loss function during training. According to the experimental results, the improved YOLOv5l has higher detection accuracy than the original YOLOv5 and is capable of detecting small targets, multiple targets, and overlapped targets with higher accuracy. It's important to note that the calculation of mAP in YOLO may vary slightly depending on the specific implementation and variations of the YOLO algorithm. Additionally, the mAP calculation considers factors like multi-object detection and handling overlapping detections to comprehensively evaluate the model's performance.

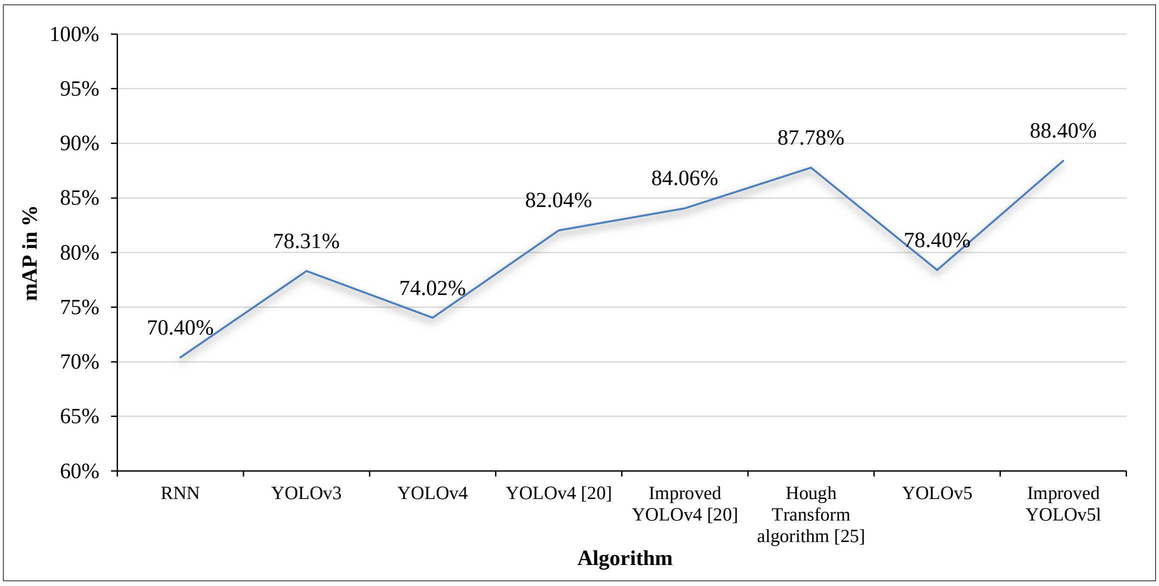


Figure 7. Accuracy comparison chart

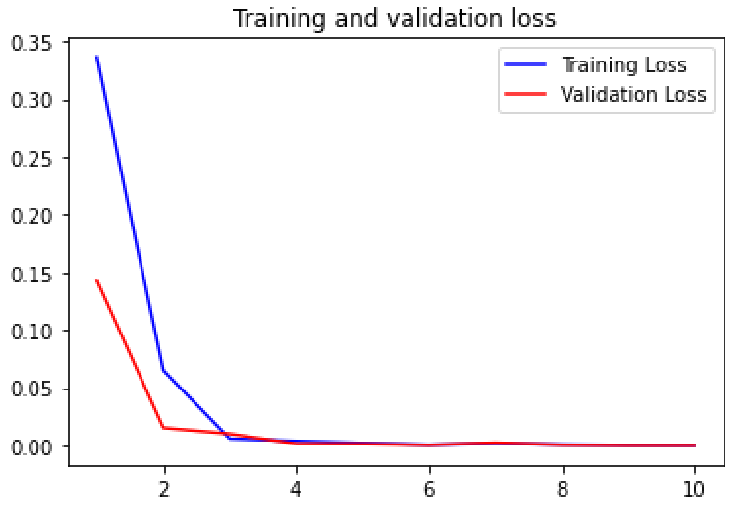


Figure 8. Training Vs Validation Loss of Recurrent neural network (RNN)

Table 3 summarises the performance metrics of various models, including the proposed model YOLOv5l, RNN, YOLOv3, YOLOv4 and Improved YOLOv4 [20]. Accuracy, precision, sensitivity, specificity, and the F score are measured. In terms of accuracy, the proposed model (YOLOv5l) outperforms the other models, obtaining an accuracy of 88.40%. It also demonstrates high precision (89.60%) and sensitivity (90.80%), indicating its ability to detect and categorize objects precisely. The proposed model outperforms YOLOv4 and the improved YOLOv4 [20] with an F score of 92.39%, a metric incorporating precision and recall. Overall, the proposed model demonstrates superior performance across multiple metrics, highlighting its efficacy in object detection.

Table 3. Treatment Values for Statistical Test

	RNN	YOLOv3	YOLOv4	Improved YOLOv4 [20]	YOLOv5l Proposed Model
Accuracy	70.40	78.30	74.02	84.06	88.40
Precision	73.47	79.54	75.13	85.76	89.60
Sensitivity	85.23	89.69	89.69	88.34	90.80
Specificity	42.86	57.14	44.86	62.34	67.14
F Score	78.92	84.31	81.77	89.90	92.39

Various statistical tests, including Friedman Aligned Rank Test, Wilcoxon Test, Quade Test, and Paired t-test, were employed to determine the presence of statistically significant differences between the means of three or more independent groups. Specifically, the article evaluated the accuracy, precision, and F-score efficiency of several classifier algorithms, namely RNN, YOLOv3, YOLOv4, and the proposed YOLOv5 model, as shown in Table 3. The impact of classification results on these four algorithms was analyzed using statistical tests. Here, Friedman Aligned Rank Test, Wilcoxon Test, Quade Test, and Paired t-test were conducted to show the impact of classification results on the above four algorithms. Table 4 shows that all the results are significant at $p < 0.05$, suggesting that the treatments differ significantly for that significance level. Table 4 shows that using improved YOLOv5l yields better performance, and the result is significant.

Table 4. Comparison of Statistical Test Performance

The Friedman Test	Wilcoxon Test	Quade Test	Paired t-Test
The X2r statistic is 13.86 (3, N = 5).	The value of z is -2.0226.	F score: 12.03	Estimated between YOLOv5 and YOLOv5l -the proposed model.
The p-value is 0.0031.	The critical value for W at N= 5 (p <0.05) is 0.	P: 0.0079	The value of t is 3.590924. The value of p is 0.02294.
The result is significant at p < 0.05	The result is significant at p< 0.05.	The result is very significant at p <0.05	The result is significant at p < 0.05.

The objects like chairs, vans, buses, dogs, cows, TV, mobile and known persons are tested by blind people using the developed mobile applications. The testing phase involved five individuals who were given 20 to 25 objects to detect, and their accuracy was measured. The accuracy of the classifiers compared to that of the individuals is presented in Table 5. Notably, the proposed work could not be directly compared to existing works because testing was conducted in real time with real-world pictures. The testing was done with face-to-face interaction and using the “Eye- A lightweight Mobile Application.” From Table 5, it is inferred that the improved YOLOv5l model produces better results, which can be attributed to incorporating L1 and L2 regularisation methods.

Table 5. Accuracy of the classifiers -tested by blind people

	RNN	YOLOv3	YOLOV4	YOLOv5l Proposed Model
Person 1	75.5	81.5	79.0	85.5
Person 2	73.4	84.2	75.13	82.0
Person 3	85.23	85.8	84.5	87.0
Person 4	75.45	83.4	80.2	81.2
Person 5	79.45	80.5	81.4	84.3

The methodologies' evaluation also includes an assessment of their performance in terms of training time, testing time and result rendering speed. We trained our model in a short span of approximately 29 minutes and 18 seconds. RNN, which was our first algorithm of choice, took approximately 4 hours for the training process, while all the YOLO models took approximately 30 to 35 minutes. The comparison graph is shown in Figure 9. The training time of YOLOv5, YOLOv4, and YOLOv5l can vary depending on several factors, such as hardware specifications, dataset size, training configurations, and the complexity of the object detection task. YOLOv5 is known for its faster training time compared to previous versions. The training time for YOLOv5 depends on factors like the dataset size, number of classes, and hardware setup. On a high-end GPU, such as an NVIDIA RTX 2080 Ti or an NVIDIA V100, training YOLOv5 on a medium-sized dataset can take several hours to a day or two. YOLOv5l is an enhanced version of YOLOv5 that offers better performance but takes slightly longer training time due to its larger model size. The training time for YOLOv5l can be similar to or slightly longer than YOLOv5, depending on the dataset size and hardware setup.

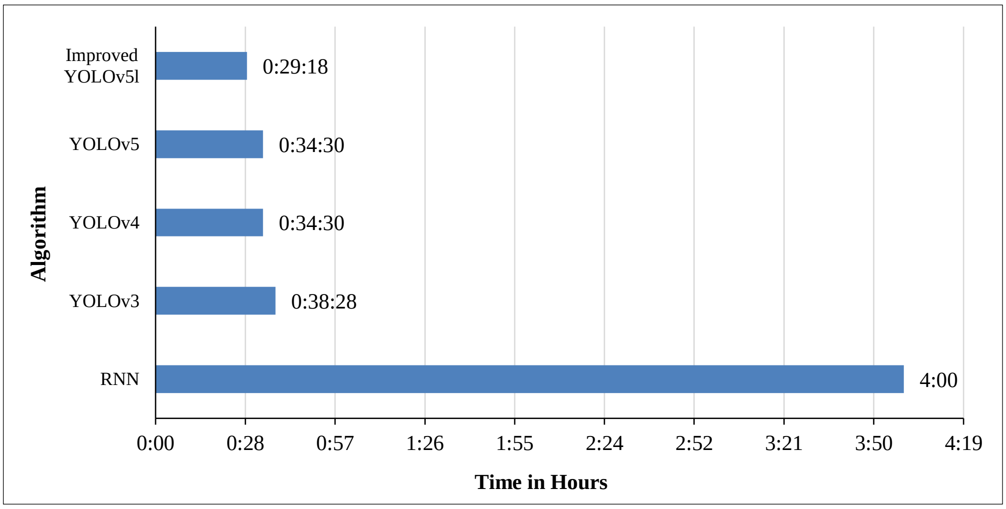


Figure 9. Training time comparison graph

After the models were trained, they underwent testing with a dataset of 10,000 images. The results indicated that our proposed model, which employed the YOLOv5l architecture, could detect objects in the testing images within 5.6 seconds, as depicted in Figure 10. It's important to note that hardware acceleration techniques like GPU utilization, optimization of the inference code, and other hardware-specific optimizations can influence the testing time. Additionally, smaller input image sizes can improve the testing time but may affect the detection accuracy, especially for small objects.

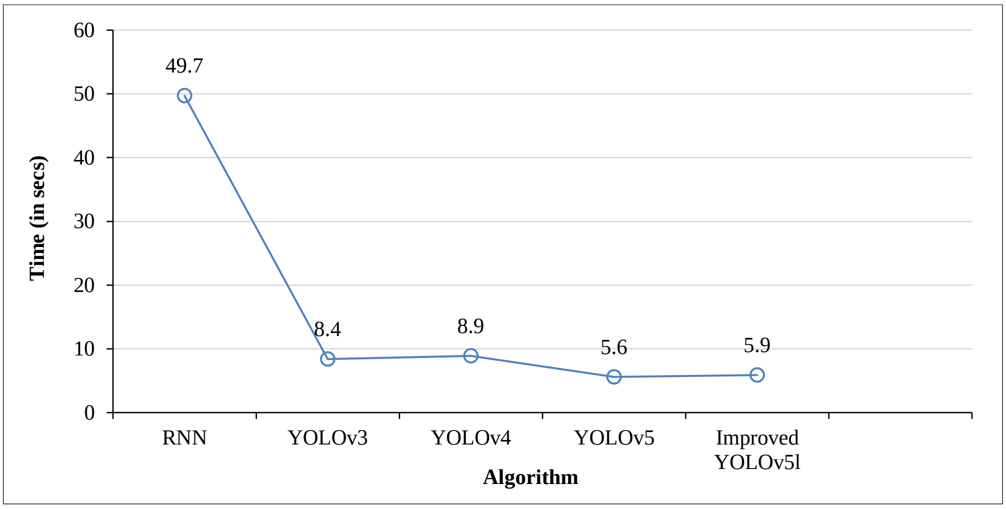


Figure 10. Testing time comparison chart

Overall, YOLOv5 generally offers faster testing time than YOLOv4, and YOLOv5l may have a slightly longer testing time due to its larger model size. The exact testing time can vary based on hardware specifications, input image size, and model complexity, and it is recommended to benchmark the models on the target hardware to get precise performance measurements.

Figure 11 shows the comparison of rendering speeds among various methodologies. Rendering speed can be influenced by the size and complexity of the input image, as well as the number and size of objects to be detected. Larger input image sizes and scenes with numerous objects can increase the rendering time. Among the evaluated methods, YOLOv3 demonstrated the rendering result at the lowest speed with an average of approximately 3 seconds, followed by YOLOv5l, which produced results in an average time of 3.1 seconds.

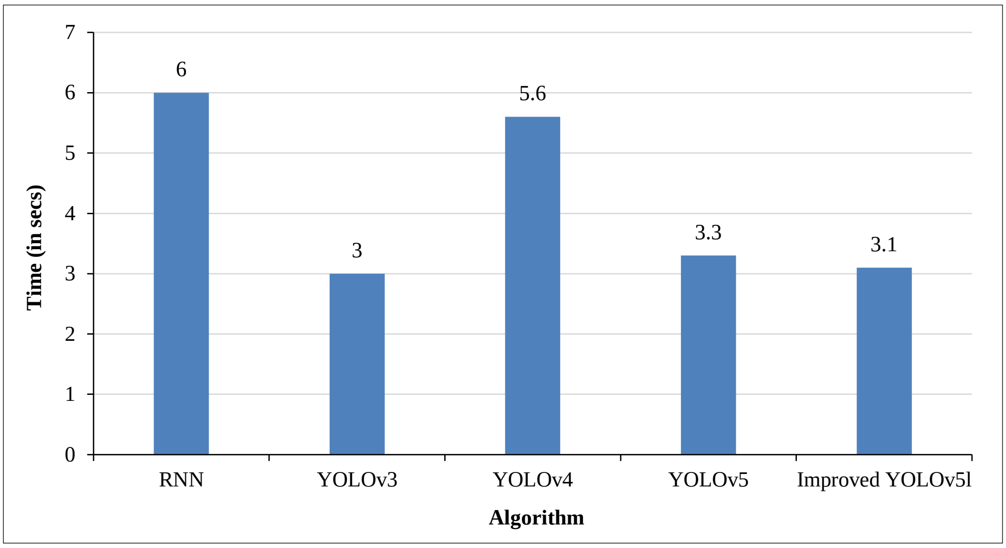


Figure 11. **Result rendering speed comparison**

5-2- Discussion

All the models were evaluated using four criteria: accuracy, training time, testing time, and result rendering speed. The final model built from the improved YOLOv5 surpassed the performance of existing models, achieving an accuracy of 88.4. While YOLOv3 has a higher rendering speed than our final model, other salient factors favour the proposed model.

A diverse and representative training set is required to use YOLOv5l effectively. It should include relevant object classes and scenarios specific to the application. YOLOv5l is also optimized for detecting objects within specific size ranges. Smaller variants of YOLOv5 or alternative object detection algorithms may be more appropriate for larger objects. Considering these factors aids in assessing the performance and suitability of YOLOv5l.

YOLOv5l requires substantial computational resources, both during training and inference. For efficient training and processing, sufficient computational resources, such as an adequate graphics card or hardware accelerators, are required. This can limit its usability in resource-constrained environments or on devices with limited computing capabilities. YOLOv5l has a larger model size than smaller variants, which can pose challenges for deployment in memory-constrained environments or on devices with limited storage capacity. Like other object detection algorithms, YOLOv5l may face challenges in accurately detecting and localizing objects in densely crowded scenes, particularly when objects heavily overlap.

YOLOv5l has a larger model size compared to smaller variants, which can be a limitation in resource-constrained environments or when deploying the model on devices with limited memory and processing capabilities. YOLOv5l, like any deep learning model, requires a large and diverse training dataset to achieve optimal performance. Collecting and annotating a large-scale dataset for training YOLOv5l can be time-consuming and resource-intensive. Moreover, as with many other object detection algorithms, accurately detecting and localizing objects in densely crowded scenes, where objects heavily overlap or occlude each other, can be challenging for YOLOv5l.

Overall, YOLOv5l is a valuable choice for object detection tasks due to its enhanced accuracy and the ability to customize the model based on resource constraints and application requirements. It provides more accurate and trustworthy object detection results than smaller variants. In addition, the adaptability of YOLOv5l enables simple scalability of the model to accommodate various deployment scenarios, whether in environments with limited resources or applications with specific requirements. By modifying the model, YOLOv5l can optimize its efficacy and efficiency for various use cases.

Usability testing was conducted on the prototype developed in this study. For testing purposes, 140 users' opinions were gathered to cover various scenarios using Google Forms, which included their opinions on the application's layout, performance, and overall rating. The results of the usability testing are presented in Figures 12 and 13. Integrating YOLO into a mobile app for usability testing requires additional development work to connect the YOLO model with the app's user interface and capture relevant data. It's also crucial to consider user privacy and data protection when implementing such features.

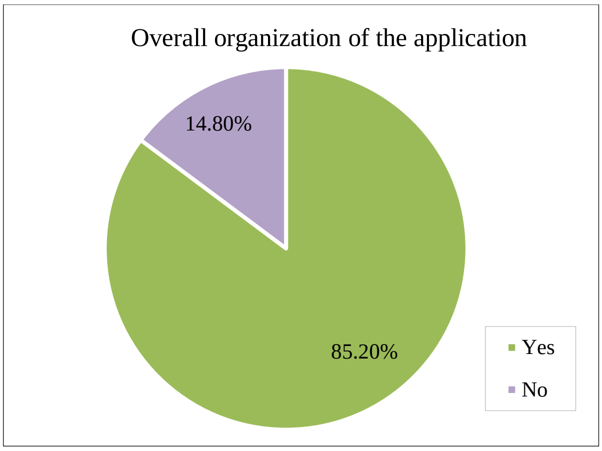


Figure 12. Feedback about the application layout

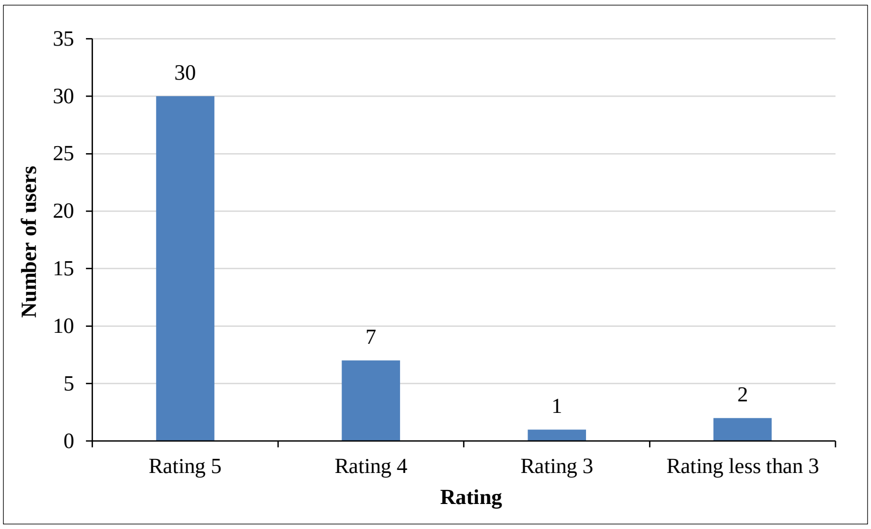


Figure 13. Rating about the performance of the application

6- Conclusion

The proposed lightweight mobile application "The Eye" is implemented using an enhanced version of the YOLO5l algorithm. The algorithm is trained and evaluated using the coco dataset and real-time images to significantly alleviate visually challenged people's daily challenges. The proposed algorithm's performance is compared to RNN, YOLOv3, YOLOv4, YOLOv5, and YOLOv5l. Four criteria were used to evaluate each model: precision, training time, testing time, and rendering performance. The proposed YOLOv5l model outperformed other object detection methods in terms of accuracy, speed, training time, and testing time. The YOLOv5l constructed from the enhanced YOLOv5 outperforms existing models and has an accuracy of 88.4%. Upon object recognition, the output is converted into voice messages using the built-in TTS class of Android, catering to the needs of visually challenged people. The developed application enables visually challenged people to effortlessly recognize their surroundings without requiring physical interaction or financial transactions. It is important to note that YOLOv5l has a larger model size than its smaller variants, which can pose limitations in resource-constrained environments or when deploying the model on devices with limited memory and processing capabilities.

The larger model size may require additional computational resources during the inference process. Future enhancements may include integrating text recognition, emotion recognition, and gender classification, providing users with a more detailed representation of the environment. The choice of post-processing techniques, such as non-maximum suppression (NMS) and thresholding, can also impact the final performance of the model. Future research can focus on designing more efficient and lightweight YOLO architectures that can achieve high accuracy while reducing computational requirements. This includes exploring model compression techniques, network architecture search, and efficient parameterization schemes to improve inference speed and reduce memory footprint. Continued research and development in this field will lead to further advancements in performance and accessibility, benefiting visually challenged people in their daily lives.

7- Declarations

7-1- Author Contributions

Conceptualization, K.S.M.A. and S.S.; methodology, K.S.M.A., S.G.B., and R.S.; validation, S.G.B., R.S., and S.K.; formal analysis, R.S., D.C., and J.K.; investigation, R.S. and J.K.; resources, S.G.B., S.S., and S.K.; writing—original draft preparation, S.G.B., D.C., and R.S.; writing—review and editing, S.S.; K.S.M.A., and S.K.; visualization, J.K.; supervision, S.S. and D.C.; project administration, K.S.M.A. All authors have read and agreed to the published version of the manuscript.

7-2- Data Availability Statement

Publicly available datasets were analyzed in this study. This data can be found in COCO Dataset [33]: https://cocodataset.org/#home.

7-3- Funding

This research is supported by Multimedia University, Malaysia.

7-4- Institutional Review Board Statement

Not applicable.

7-5- Informed Consent Statement

Not applicable.

7-6- Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this manuscript. In addition, the ethical issues, including plagiarism, informed consent, misconduct, data fabrication and/or falsification, double publication and/or submission, and redundancies have been completely observed by the authors.

8- References

[1] WHO. (2023). Blindness and vision impairment. World Health Organization (WHO), Geneva, Switzerland. Available online: https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment (accessed on April 2023).

[2] Fu, X. (2021). Mobile assistant app for visually impaired people, with face detection, gender classification and sound representation of image. Electrical Engineering Department, Stanford University, California, United States.

[3] Mao, Q.-C., Sun, H.-M., Liu, Y.-B., & Jia, R.-S. (2019). Mini-YOLOv3: Real-Time Object Detector for Embedded Applications. IEEE Access, 7, 133529–133538. doi:10.1109/access.2019.2941547.

[4] Statista (2023). Number of smartphone mobile network subscriptions worldwide from 2016 to 2022, with forecasts from 2023 to 2028. Statista Inc, New York, United States. Available online: https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide (Accessed on June 2023).

[5] Lara, S.D., & Labrador, M. A. (2012). A mobile platform for real-time human activity recognition. 2012 IEEE Consumer Communications and Networking Conference (CCNC). doi:10.1109/ccnc.2012.6181018.

[6] Huu, P. N., & Xuan, K. D. (2021). Proposing Algorithm Using YOLOV4 and VGG-16 for Smart-Education. Applied Computational Intelligence and Soft Computing, 2021. doi:10.1155/2021/1682395.

[7] Wu, L., Ma, J., Zhao, Y., & Liu, H. (2021). Apple detection in complex scene using the improved yolov4 model. Agronomy, 11(3). doi:10.3390/agronomy11030476.

[8] Li, Y., Wang, H., Dang, L. M., Nguyen, T. N., Han, D., Lee, A., Jang, I., & Moon, H. (2020). A deep learning-based hybrid framework for object detection and recognition in autonomous driving. IEEE Access, 8, 194228–194239. doi:10.1109/ACCESS.2020.3033289.

[9] Thammarak, K., Sirisathitkul, Y., Kongkla, P., & Intakosum, S. (2022). Automated Data Digitization System for Vehicle Registration Certificates Using Google Cloud Vision API. Civil Engineering Journal, 8(7), 1447-1458. doi:10.28991/CEJ-2022-08-07-09.

[10] Hu, X., Wei, Z., & Zhou, W. (2021). A video streaming vehicle detection algorithm based on YOLOv4. 2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC). doi:10.1109/iaeac50856.2021.9390613.

[11] Kurdthongmee, W., Kurdthongmee, P., Suwannarat, K., & Kiplagat, J. K. (2022). A YOLO Detector Providing Fast and Accurate Pupil Center Estimation using Regions Surrounding a Pupil. Emerging Science Journal, 6(5), 985-997. doi:10.28991/ESJ-2022-06-05-05.

[12] 25-Linglin, H., Qiang, L., Xianzhen, H., & Maosong, L. (2020). Research on pruning algorithm of target detection model with YOLOv4. 2020 Chinese Automation Congress (CAC). doi:10.1109/cac51589.2020.9326798.

[13] Anbananthen, K. S. M., Subbiah, S., Chelliah, D., Sivakumar, P., Somasundaram, V., Velshankar, K. H., & Khan, M. K. A. A. (2021). An intelligent decision support system for crop yield prediction using hybrid machine learning algorithms. F1000Research, 10(1143). doi:10.12688/f1000research.73009.1.

[14] Subbiah, S., Anbananthen, K. S. M., Thangaraj, S., Kannan, S., & Chelliah, D. (2022). Intrusion detection technique in wireless sensor network using grid search random forest with Boruta feature selection algorithm. Journal of Communications and Networks, 24(2), 264–273. doi:10.23919/jcn.2022.000002.

[15] Sridevi, S., Parthasarathy, S., & Rajaram, S. (2018). An effective prediction system for time series data using pattern matching algorithms. International Journal of Industrial Engineering: Theory Applications and Practice, 25(2), 123–136. doi:10.23055/ijietap.2018.25.2.3318.

[16] Wu, D., Lv, S., Jiang, M., & Song, H. (2020). Using channel pruning-based YOLO v4 deep learning algorithm for the real-time and accurate detection of apple flowers in natural environments. Computers and Electronics in Agriculture, 178, 105742. doi:10.1016/j.compag.2020.105742.

[17] Abdurahman, F., Fante, K. A., & Aliy, M. (2021). Malaria parasite detection in thick blood smear microscopic images using modified YOLOV3 and YOLOV4 models. BMC Bioinformatics, 22(1). doi:10.1186/s12859-021-04036-4.

[18] Albahli, S., Nida, N., Irtaza, A., Yousaf, M. H., & Mahmood, M. T. (2020). Melanoma Lesion Detection and Segmentation Using YOLOv4-DarkNet and Active Contour. IEEE Access, 8, 198403–198414. doi:10.1109/ACCESS.2020.3035345.

[19] Yilmazer, R., & Birant, D. (2021). Shelf auditing based on image classification using semi-supervised deep learning to increase on-shelf availability in grocery stores. Sensors (Switzerland), 21(2), 327. doi:10.3390/s21020327.

[20] Fu, H., Song, G., & Wang, Y. (2021). Improved yolov4 marine target detection combined with CBAM. Symmetry, 13(4). doi:10.3390/sym13040623.

[21] Parico, A. I. B., & Ahamed, T. (2021). Real time pear fruit detection and counting using yolov4 models and deep sort. Sensors, 21(14). doi:10.3390/s21144803.

[22] Anbananthen, S. K., Sainarayanan, G., Chekima, A., & Teo, J. (2006). Data Mining using Pruned Artificial Neural Network Tree (ANNT). 2nd International Conference on Information & Communication Technologies, 16 October 2006, Damascus, Syria. doi:10.1109/ictta.2006.1684577.

[23] Cai, Y., Luan, T., Gao, H., Wang, H., Chen, L., Li, Y., Sotelo, M. A., & Li, Z. (2021). YOLOv4-5D: An Effective and Efficient Object Detector for Autonomous Driving. IEEE Transactions on Instrumentation and Measurement, 70, 1–13. doi:10.1109/tim.2021.3065438.

[24] Ponnada, S., Sekharamantry, P. K., Dayal, A., Yarramalle, S., Vadaparthi, N., & Hemanth, J. (2021). An assisting model for the visually challenged to detect bus door accurately. Telkomnika (Telecommunication Computing Electronics and Control), 19(6), 1924–1934. doi:10.12928/TELKOMNIKA.v19i6.19811.

[25] Bagyam, M. L. N., Indujha, S., Karthika, P., & Hariharan, T. (2021). Smart hearing and visually impaired passenger voice alert system. AIP Conference Proceedings. doi:10.1063/5.0068999.

[26] Kumar, L. A., Renuka, D. K., Rose, S. L., & Wartana, I. M. (2022). Deep learning based assistive technology on audio visual speech recognition for hearing impaired. International Journal of Cognitive Computing in Engineering, 3, 24-30. doi:10.1016/j.ijcce.2022.01.003.

[27] Ranjan, A., & Navamani, T. M. (2019). Android-Based Blind Learning Application. Ambient Communications and Computer Systems. Advances in Intelligent Systems and Computing, vol 904. Springer, Singapore. doi:10.1007/978-981-13-5934-7_22.

[28] 30-Arvind Bhile, A., & Hole, V. (2020). Real-Time Environment Description Application for Visually Challenged People. Second International Conference on Computer Networks and Communication Technologies. ICCNCT 2019. Lecture Notes on Data Engineering and Communications Technologies, 44, Springer, Cham, Switzerland. doi:10.1007/978-3-030-37051-0_38.

[29] Bhatia, S., Devi, A., Alsuwailem, R. I., & Mashat, A. (2022). Convolutional Neural Network Based Real Time Arabic Speech Recognition to Arabic Braille for Hearing and Visually Impaired. Frontiers in Public Health, 10. doi:10.3389/fpubh.2022.898355.

[30] Baskar, A., Kumar, T. G., & Samiappan, S. (2023). A vision system to assist visually challenged people for face recognition using multi-task cascaded convolutional neural network (MTCNN) and local binary pattern (LBP). Journal of Ambient Intelligence and Humanized Computing, 14(4), 4329–4341. doi:10.1007/s12652-023-04542-8.

[31] Al-Allaf, A. F., & Rida, M. M. (2023). Design and implementation of a walking stick aid for visually challenged people. AIP Conference Proceedings. doi:10.1063/5.0116712.

[32] Khan Shishir, Md. A., Rashid Fahim, S., Habib, F. M., & Farah, T. (2019). Eye Assistant: Using mobile application to help the visually impaired. 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT). doi:10.1109/icasert.2019.8934448.

[33] COCO. (2023). Common Objects in Context. Available online: https://cocodataset.org/#home (accessed on April 2023).