Türk Bilim ve Mühendislik Dergisi, 6(2): 72-81, 2024

Araştırma Makalesi / Research Article

Türk Bilim ve Mühendislik Dergisi
Turkish Journal of Science and Engineering

www.dergipark.org.tr/tjse

Cross-Assist: Road Assistance Application for Visually Impaired People

Ayşe Demirhan[1*], Dilruba Alkan[2]

[1]Gazi Üniversitesi, Teknoloji Fakültesi, Elektrik-Elektronik Mühendisliği Bölümü – Ankara-Türkiye

***Sorumlu yazar: ayseoguz@gazi.edu.tr**

ARTICLE INFO

Received: 04/03/2024
Accepted: 27/09/2024

Keywords: *Assistive technology, Crosswalk detection, Pedestrian traffic light detection, Visual impairment, YOLO object detector*

DOI: 10.55979/tjse.1447019

ABSTRACT

According to WHO (World Health Organization) 2.2 billion people in the world have visual impairment. About 40 million of them experience complete vision loss. This number is substantial for the world population. Lack of visual function is one factor that makes it difficult for the individual to participate in social life. Because a barrier-free life is aimed, studies have emerged due to the difficulties encountered. One of these difficulties is that they need help seeing pedestrian lights and roads to cross the street. In this study, a mobile application is designed to address this issue. The application provides visually impaired individuals with voice alerts about the status of crosswalks and traffic lights. This mobile application was developed using Flutter. The convolutional neural network model and YOLO (You Only Look Once) v2Tiny algorithm were used for real-time object recognition from the images taken from the mobile phone camera. Mobile application successfully recognizes red light, green light, and crosswalk with 89.52%, 89.1%, and 88.57% accuracies, respectively. The novelty of this study lies in incorporating both pedestrian traffic light detection and crosswalk identification within a mobile application.

Cross-Assist: Görme Engelli Kişiler için Yol Yardım Uygulaması

MAKALE BİLGİSİ

Alınış tarihi: 04/03/2024
Kabul tarihi: 27/09/2024

Anahtar Kelimeler: *Destekleyici teknoloji, Yaya geçidi algılama, Yaya trafik ışıkları algılama, Görme engelli, YOLO nesne algılayıcı*

DOI: 10.55979/tjse.1447019

ÖZET

DSÖ'ye (Dünya Sağlık Örgütü) göre dünyada 2.2 milyar kişinin görme engeli bulunmaktadır. Bu kişilerden yaklaşık 40 milyonu tamamen görme kaybı yaşamaktadır. Bu sayı dünya nüfusu için önemli bir rakamdır. Görme fonksiyonunun eksikliği, bireyin sosyal yaşama katılımını zorlaştıran bir faktördür. Engelsiz bir yaşam hedeflendiği için karşılaşılan zorluklar nedeniyle birçok çalışma ortaya çıkmıştır. Bu zorluklardan biri, görme engelli bireylerin yolda karşıya geçerken yaya ışıklarını ve yolları görmelerine yardımcı olmaya ihtiyaç duymalarıdır. Bu çalışmada bu soruna çözüm bulmak amacıyla tasarlanmış bir mobil uygulama geliştirilmiştir. Uygulama, görme engelli bireylere yaya yollarının ve trafik ışıklarının durumu hakkında sesli uyarılar sağlamaktadır. Bu mobil uygulama, Flutter kullanılarak geliştirilmiştir. Mobil telefon kamerasından alınan görüntüler üzerinden gerçek zamanlı nesne tanıma için konvolüsyonel sinir ağı modeli ve YOLO (You Only Look Once) v2 Tiny algoritması kullanılmıştır. Mobil uygulama, kırmızı ışık, yeşil ışık ve yaya geçidi tanımayı sırasıyla %89.52, %89.1 ve %88.57 doğruluk oranlarıyla başarıyla gerçekleştirmektedir. Bu çalışmanın yeniliği, bir mobil uygulama içinde hem yaya trafik ışığı tespiti hem de yaya geçidi tanımlamasını içermesidir.

1. Introduction

Various methods and technologies have been developed to enable visually impaired individuals to travel safely and independently in traffic. Among these methods, traditional tools such as walking canes and trained guide dogs play a significant role. Additionally, equipping traffic signals with auditory alerts is an effective approach to help visually impaired individuals perceive environmental conditions and manage traffic movements. Mobile applications also play a crucial role in this domain, offering features such as location-based auditory guidance and alerting users to environmental hazards, thus assisting visually impaired individuals in safe navigation. Voice map applications describe streets and key locations audibly, while sensors embedded in clothing and equipment detect environmental obstacles and provide users with vibration or auditory warnings. Each of these methods offers different advantages in enhancing the safety of visually impaired individuals in traffic, and it is essential to select options tailored to individuals' needs and preferences.

Mobile applications developed for visually impaired individuals are designed to reduce daily barriers and enhance independence. These applications typically offer various features such as location-based auditory guidance, voice reading of texts, color and light detection, navigation support, and access to online shopping or social media platforms. Continuously evolving, these technologies aim to improve users' quality of life by enabling greater independence in daily activities for visually impaired individuals.

Most of the time, artificial intelligence is utilized in the development of these technologies. Artificial intelligence aims to adapt human actions to machines and systems by simulating human intelligence and behavior. One of the human characteristics aimed to adapt is visual function. Machine vision is a set of events consisting of object detection, sensing, and following the objects performed by robots, machines, etc. The foundation of machine vision involves converting captured images into digital data using advanced processing techniques. This conversion is achieved through artificial neural networks, which are essential for transforming visual information into a digital format. These neural networks are vital to the operation of machine learning and deep learning systems, providing the primary inputs that allow these technologies to analyze and interpret visual data. The model obtained from training an artificial neural network ensures that a machine or system can accurately predict outcomes, recognize patterns, classify data, and perform specific tasks by learning from and adapting to the data it has been trained on.

Machine vision finds applications across various fields including industrial production processes, quality control, traffic management, and healthcare, among others. Its continuous advancements not only benefit industries but also contribute to simplifying our daily lives. The adaptability and wide-ranging applicability of machine vision make it a valuable tool for solving diverse problems.

By mimicking human visual functions, machine vision technology has been developed to address a multitude of challenges. It has also been harnessed to enhance the lives of visually impaired individuals. One specialized area within machine vision focuses on addressing the difficulties faced by visually impaired pedestrians in navigating traffic. Some studies in this domain involve processing images of traffic lights, while others focus on detecting crosswalks (Li et al., 2020; Shangguan et al., 2014).

Several mobile applications have been developed as part of these studies, but most of them specialize in either traffic light recognition or crosswalk detection, not both.

The contribution of this study is that it includes both the identification of pedestrian traffic lights and crosswalks in a mobile application. For this purpose, a mobile application has been developed that recognizes pedestrian traffic lights and the crosswalk and uses sound to inform visually impaired people whether they should cross the road. This mobile application aims to minimize the difficulties that visually impaired people encounter in traffic and overcome some challenges in joining social life.

Convolutional neural network models are widely employed for machine vision tasks (Khan et al., 2018; Sinha et al., 2018; Srinivas et al., 2016). In this study, YOLO, a highly successful algorithm for real-time object identification (Huang et al., 2018) is utilized. YOLO stands out due to its speed and real-time object identification capabilities compared to other algorithms. For this study, images sourced from the internet and mobile phone cameras constitute the dataset used to train and test YOLO. The resulting weight files from this training process are converted into the necessary formats and integrated into the mobile application.

2. Related Work

In complex urban traffic conditions, blind individuals typically rely on traditional walking sticks for navigation, using the tip of the stick to touch the ground and sweep from side to side to detect obstacles and gather information about the road ahead. However, these canes struggle to detect distant objects or obstacles that are elevated above ground level. To address these limitations, artificial intelligence-based solutions for obstacle avoidance have been developed, utilizing hardware such as smart glasses, smartphones, Raspberry Pi, ultrasonic sensors, water sensors, cameras, Arduino, belts, harnesses, and bone conduction headsets (Wang et al., 2023).

Numerous studies have been conducted employing machine vision techniques to assist visually impaired individuals. These studies can be categorized into three primary domains (Cheng & Tsai, 2024; Kuriakose et al., 2023, Hwang et al., 2024, J., Ash et al., 2018; Ghilardi et al., 2018; Cheng et al., 2018; Rajwani et al., 2018; Tosun & Karaarslan, 2018; Dionisi et al., 2012; Arora et al., 2019). The first domain pertains to the context in which the study is conducted, with the type of objects to be identified determining the focus of such investigations. Another significant aspect is the methodology employed, with widely utilized real-time image processing techniques including R-CNN (Region-based Convolutional Neural Network), Faster R-CNN, SSD (Single Shot Detector), and YOLO (Li et al., 2020; Shangguan et al., 2014). The third major differentiator among these studies is the specific environment in which the research is conducted. In some cases, the trained model is embedded in a mobile application, while sometimes it can be used with a hardware module containing a camera (Li et al., 2020; Ash et al., 2018; Cheng et al., 2018; Mahesh et al., 2021; Son & Weiland, 2022; Cheng et al., 2017).

In the study of Li et al (2020), a system named Cross-Safe is designed to offer accurate and accessible guidance to visually impaired individuals crossing intersections, integrated into a comprehensive smart wearable device. They were addressing the red-light-green-light, go-no-go

problem, due to the significant lack of accessible pedestrian signals in New York City's urban infrastructure. Cross-Safe utilizes CNN based deep learning techniques for real-time detection and recognition of pedestrian signals. Their system needs extra hardware equipment such as Nvidia Jetson TX2, bone-conduction headset for voice feedback and power bank and its accessories to operate (Li et al., 2020).

Son & Weiland (2022) proposed a wearable system to assist visually impaired users at signalized crosswalks. The system provides verbal instructions for the user to either move "forward" or adjust their direction by veering left or right. The navigation system operates on a commercially available mobile computer, and includes hardware components such as an RGB-D camera, a BNO055 IMU sensor, and bone conduction headphones for user interaction. It also utilizes a pre-mapped layout of the crosswalks. Each software component operates on the Robot Operating System (ROS). The system uses the location of a crosswalk end plate to link the pre-mapped layout, collected with LiDAR (Light Detection and Ranging), to real-time RGB-D streaming data. A modified U-net structure and images from 16 crosswalks near their research lab were employed in the system (Son & Weiland, 2022).

In several studies, real-time object detection for pedestrian traffic light detection was conducted using R-CNN, Faster R-CNN, SSD, and YOLO (Ash et al., 2018; Ghilardi et al., 2018). Upon examining these studies, it is evident that some utilized a portable device rather than a mobile application (Li et al., 2020; Cheng et al., 2018). Additionally, the camera position varied across these prior studies.

Kuriakose et al. (2023) introduced a smartphone-based navigation assistant that leverages deep learning to provide users with detailed information about obstacles, including their type, position, distance, motion status, and surrounding scene details. The system primarily consists of a smartphone and a bone conduction headset, along with several software modules for obstacle detection, distance and position estimation, motion detection, and scene recognition. Navigation information is delivered to the user through the bone conduction headset. For obstacle detection, they employed a lightweight model, EfficientDet-Lite4, from the Efficient Det family. The system can recognize various objects from both indoor and outdoor environments, such as cars, chairs, people, plants, stop signs, traffic lights, and trees (Kuriakose et al., 2023).

Cheng & Tsai (2024) utilized image processing techniques to pinpoint the central position of crosswalks. Working within both RGB and HSV color spaces, they first removed shadows from the crosswalk images and then extracted the white stripes. By determining the middle line of the crosswalk, they provided guidance to visually impaired individuals for safely crossing the road. Once the center line was identified, directional signals were delivered through a wearable device equipped with vibrating wristbands to assist the visually impaired in navigation (Cheng & Tsai, 2024).

Hwang et al. (2024) developed an end-to-end framework that analyzes street scene images to produce interpretable safety risk assessments for crossing streets. They gathered data on crosswalk intersections using multiview egocentric images captured by a quadruped robot, and annotated these images with safety scores based on a predefined categorization. To assess street-crossing risks, they used images labeled with safety risk levels in combination with GPT-4V, a large language model. The robot’s multiview egocentric images provided visual data, including object detection bounding boxes, segmentation masks, and optical flow. This visual information, along with text prompts, was processed by the LLM to generate both safety scores and scene descriptions (Hwang et al., 2024).

In this study, images are captured using a mobile phone’s camera. Although the camera position can vary, the plan is to hang the phone around the individual's neck to position the camera accordingly. In other studies, where images are captured from a camera on a portable device rather than a phone camera, the camera is typically positioned on wearable technologies such as a walking stick for the disabled (Shangguan et al., 2014), glasses (Cheng et al., 2018; Cheng et al., 2017), or a vest (Li et al., 2020).

In this study, crosswalks and traffic lights are simultaneously identified as different classes. The reviewed studies did not reveal any mobile applications that classify crosswalks and traffic lights concurrently. Most research in this area has been conducted using a portable camera module. In contrast, this study does not require additional hardware to identify crosswalks and traffic lights, thereby relieving the user of the need to carry an external device. Data from the mobile phone’s rear camera is processed, and the user is provided with voice guidance.

For this study, a mobile application has been developed utilizing the YOLOv2 Tiny algorithm. YOLOv2 Tiny was specifically selected due to its seamless integration with the Flutter SDK (Software Development Kit) commonly employed in mobile app development, as well as its compact size within the app. This mobile app is designed to identify pedestrian traffic lights and crosswalks simultaneously. Unlike previous related studies in the literature that focus solely on either pedestrian traffic lights or crosswalks, this app covers both aspects (Li et al., 2020; Cheng et al., 2018; Son & Weiland, 2022; Cheng et al., 2017).

3. Material ve Method

3.1. YOLO

YOLO is a deep learning algorithm, an acronym for 'You Only Look Once' and is named upon its ability to capture images at first glance. YOLO uses CNN and it is a popular algorithm in real-time object detection. It is prominent among other algorithms because it is very fast

at real-time object detection. In YOLO, the incoming image passes through the convolutional neural networks only once. This is the reason behind being fast, as other algorithms often use the images many times through the network. Also, YOLO can simultaneously identify the objects and their coordinates (Huang et al., 2018).

3.2. Dataset

Dataset size and quality play crucial roles in deep learning research. As a result, a variety of environments, lighting conditions, object dimensions, and distances in captured images are utilized to construct a more resilient dataset. The dataset used in this study comprises pedestrian traffic light and crosswalk images sourced from the Traffic Light Detection Dataset (Kaggle, 2022), the Crosswalk Dataset (Kaggle, 2020), and images collected through web searches. The Traffic Light Detection Dataset includes 2600 artificially labeled images with traffic light categories and color labels, covering nine categories: motor vehicle signal light, non-motor vehicle signal lights, left turn non-motor vehicle signal light, crosswalk signal light, lane lights, direction indicator light, flashing warning light, crossing signal light, and U-turn signal light (Kaggle, 2022). The Crosswalk Dataset is divided into four classes, each representing a different perspective. The first class consists of crosswalks viewed from the front, the second and third classes show half-lanes viewed from the left or right, and the fourth class contains non-crosswalk images, such as asphalt, passing cars, and sidewalks. These images were extracted from several videos recorded at a resolution of 1280x720 at 30 FPS in Fortaleza-CE, Brazil, during daylight hours. The images retain the same resolution as the videos, 1280x720 (Kaggle, 2020).

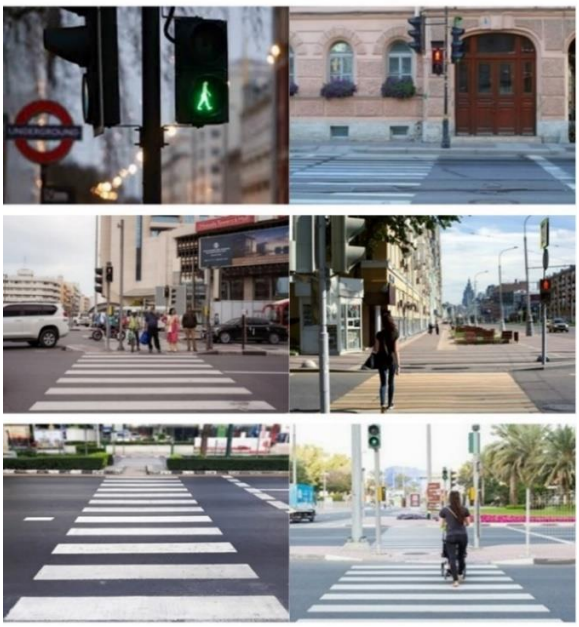


Figure 1. Example images from our image dataset

Images within the dataset are annotated for model training purposes. During annotation, the dataset is categorized into 5 distinct classes: negative, positive, green, red, and crosswalk. The positive class pertains to pedestrian traffic lights, whereas the negative class relates to vehicle traffic lights. The creation of these classes is intended to prevent the mobile application from directing pedestrians based on vehicle traffic lights. Additionally, red and green light classes are established to denote the color of pedestrian lights, while the crosswalk class is designated for identifying crosswalks. The dataset encompassing these classes comprises a total of 3367 images, with 30% allocated for testing and 70% for training the model. Specifically, the training dataset contains 2357 images, while the testing dataset comprises 1010 images. Figure 1 showcases sample images extracted from the dataset.

3.2. Real-Time Object Detection

A literature review was conducted on real-time object detection, comparing models employing various algorithms like SSD, YOLO versions, R-CNN, Faster R-CNN, etc. Among these, YOLO is acknowledged as the fastest and most accurate algorithm for real-time object detection, as evidenced by studies (Ash et al., 2018; Ghilardi et al., 2018; Francies et al., 2022; Li et al., 2022).

The YOLO algorithm has numerous versions, each tailored for specific application purposes. In this study, we tested the YOLOv4, YOLOv2, and YOLOv2-Tiny algorithms using our created dataset. Results revealed that YOLOv4 achieved the highest accuracy among these versions, while YOLOv2 exhibited the lowest accuracy in these evaluations. The YOLOv2-Tiny version was tested because the weight files generated from training with the YOLOv4 and YOLOv2 algorithms are too large for practical use in a mobile application. The smaller size of the YOLOv2-Tiny weights makes it more suitable for the limited storage and processing capabilities of mobile devices, ensuring efficient and effective performance (Chen et al., 2022). The YOLOv2-Tiny algorithm was chosen as the ultimate selection due to its compatibility with Flutter, the Software Development Kit utilized for mobile application development, its speed compared to other algorithms, and the smaller size of its output weight files.

The network architecture of YOLO (Figure 2) is similar to GoogLeNET, which consists of 22 layers of convolutional neural networks developed by Google researchers (Redmon et al., 2016). The YOLO neural network is composed of 24 convolutional layers, followed by 2 fully connected layers that define box coordinates and probabilities. To reduce the depth of characteristic maps, YOLO uses 1x1 convolutional layers on the initial layers of GoogLeNET. Initially trained on 224x224 pixel images, the input image is later scaled up to 448x448 pixels for increased prediction accuracy (Redmon et al., 2016).

Despite variations in network architectures across different YOLO versions, the fundamental structure of convolutional neural networks remains consistent. Notably, YOLO Tiny features 9 convolutional neural network layers, setting it apart from other versions. Although the accuracy of the YOLOv2-Tiny model is not the highest, achieving around 57.1% mAP (mean Average

Precision), it allows the model to run faster and take up less space, with a file size of approximately 44 MB (Wang et al., 2021).

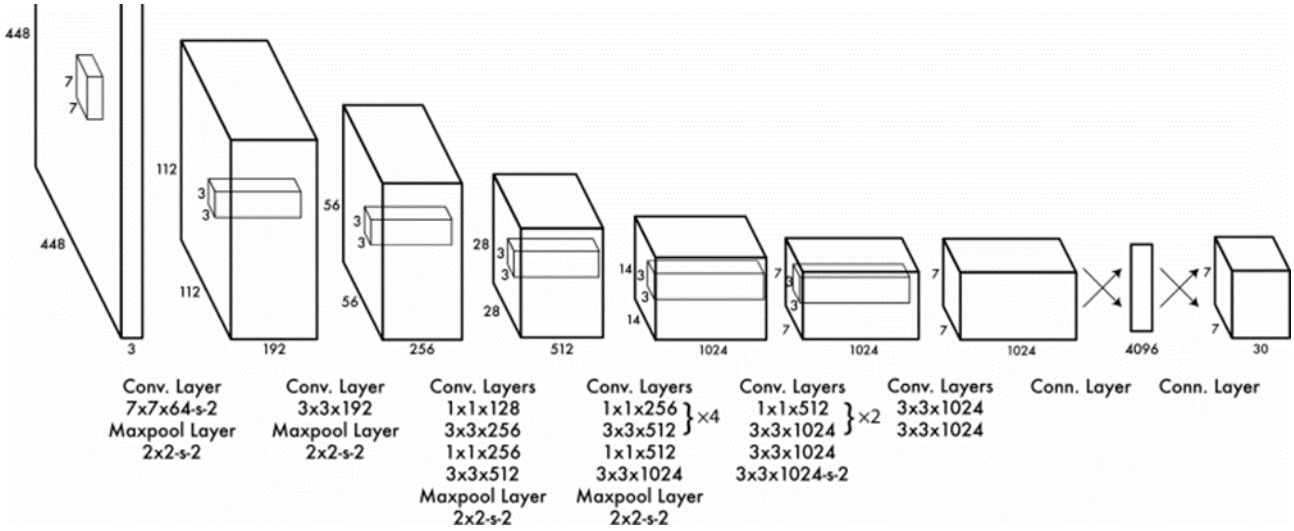


Figure 2. YOLOv2-Tiny Architecture (Redmon et al., 2016)

Table 1. Configuration Parameters

Parameter	Value
Momentum	0.9
Learning Rate	0.001
Epochs	8000, 9000
Batch	64
Subdivisions	2

The model training configuration parameters are outlined in Table 1. These parameters were established through a review of prior studies and experimentation with varying parameter values. The configuration parameters presented in Table 1 yielded optimal accuracy at the conclusion of model training for this study.

- Momentum: How much the history affects the further change of weights.
- Learning Rate: The learning rate is a hyper-parameter that controls how much to change the model in response to the estimated error each time the model weights are updated.
- Epochs: Number of iterations
- Batch: Represents the number of training examples utilized in one iteration.
- Subdivisions: The number of blocks subdivided batch.

YOLO classification loss is calculated using Eq. (1) (Redmon et al., 2016).

$$\sum_{i=0}^{s^2} 1_i^{obj} \sum_{c \epsilon classes} (p_i(c) - (\hat{p}(c))^2 \qquad (1)$$

Here's what each term in the equation represents:

- S^2: The number of grid cells in the final output feature map.
- 1_i^{obj}: An indicator function that evaluates if grid cell i contains an object or not. It equals 1 if an object is present in cell i and 0 otherwise.
- $classes$: The set of all classes that the model can predict.
- $p_i(c)$: The predicted probability of class c for grid cell i.
- $\hat{p}(c)$: The true probability of class c for the detected object.

The equation calculates the classification loss by summing the squared differences between the predicted probabilities $p_i(c)$ and the true probabilities $\hat{p}(c)$ for each class c across all grid cells i where an object is present.

The localization loss (errors between the predicted boundary box and the ground truth) is calculated using Eq. (2).

$$\lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^{B} 1_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] +$$
$$\lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i}\right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i}\right)^2 \right] \qquad (2)$$

Terms in the equation represent:

- λ_{coord}: A scaling factor that balances the impact of localization loss.
- B: The number of bounding boxes predicted per grid cell.
- 1_{ij}^{obj}: An indicator function that evaluates if bounding box j in cell i is responsible for object detection. It equals 1 if the box is responsible and 0 otherwise.

- x_i, y_i, w_i, h_i The predicted coordinates and dimensions of the bounding box in cell i.
- $\hat{x}_i, \hat{y}_i, \hat{w}_i, \hat{h}_i$: The true coordinates and dimensions of the bounding box in cell i.

Here, 1_{ij}^{obj} = 1 if the j^{th} boundary box in cell i is responsible for detecting the object, otherwise it equals to 0. λ_{coord} increases the weight for the loss in the boundary box coordinates.

The confidence loss (the objectness of the box) is calculated using Eq. (3) if an object is detected in the box.

$$\sum_{i=0}^{s^2} \sum_{i=0}^{B} 1_{ij}^{obj} (C_i - \hat{C}_i)^2 \quad (3)$$

Here, $\hat{C}_i$ is the box confidence score of the box j in cell i. 1_{ij}^{obj} if the j^{th} boundary box in cell i is responsible for detecting the object; otherwise it equals 0.

The confidence loss is calculated using Eq. (4) if an object is not detected in the box;

$$\lambda_{noobj} \sum_{i=0}^{s^2} \sum_{i=0}^{B} 1_{ij}^{obj} (C_i - \hat{C}_i)^2 \quad (4)$$

Here, 1_{ij}^{noobj} is the complement of 1_{ij}^{obj}, $\hat{C}_i$ is the box confidence score of the box j in cell i. λ_{noobj} weights down the loss when detecting background.

The final loss adds localization, confidence, and classification losses together.

Irrespective of the chosen image processing methodology, handling each pixel and conducting model training is a protracted and time-consuming endeavor. To address this issue, the parallel programming approach has emerged, enabling the execution of multiple operations concurrently. Through parallel programming, simultaneous operations can be conducted on each pixel, thereby abbreviating the training duration (Aydın et al., 2020). Processors and GPUs (Graphics Processing Units) employed in parallel programming have undergone significant advancements and have come to dominate this domain. The augmentation in GPU core numbers has rendered GPUs more advantageous than CPUs, as each core can process data simultaneously and in parallel with others. Consequently, the escalation in core numbers augments the number of processes. In this study, parallel programming is executed on the CUDA (Compute Unified Device Architecture) platform using the GPU of virtual computers on Google Colaboratory during the image processing phase.

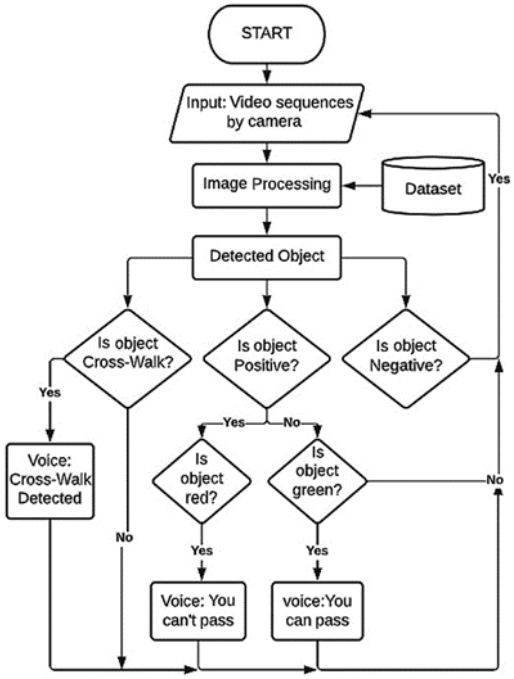


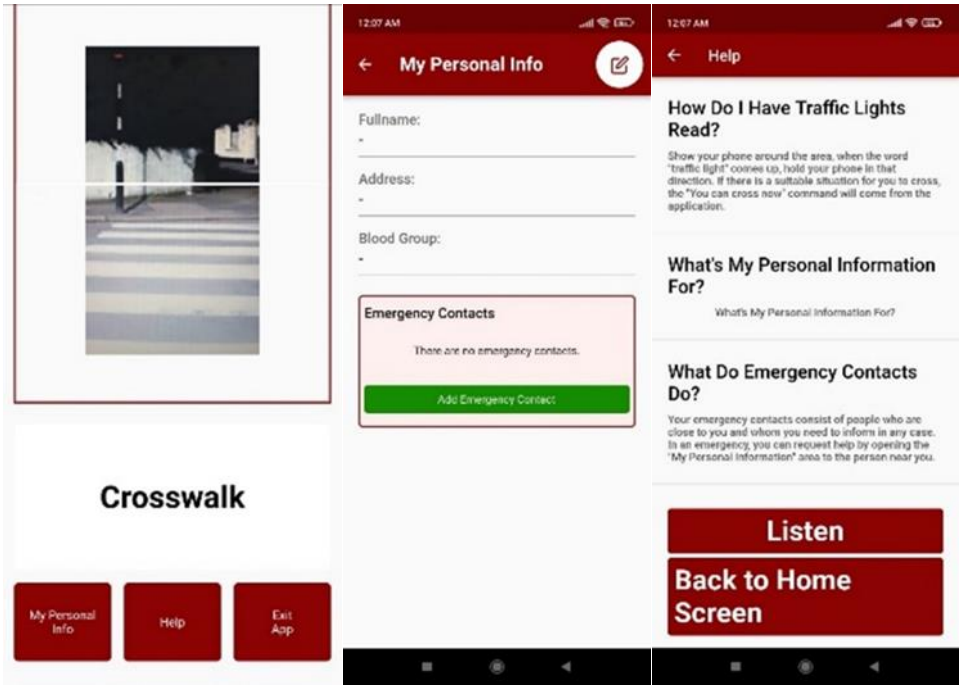
Figure 3. Flowchart of mobile application

3.4. Mobile Application

Figure 3 shows the flowchart of the mobile application. Upon opening the application, it initiates image capture from the rear camera, processes the incoming image through the filter of the trained model, and classifies real-time captures. This classification creates positive and negative classes to distinguish between pedestrian and vehicle traffic lights, thereby training the model accordingly. Consequently, only information regarding the green or red class of pedestrian traffic lights is extracted, and an audio description is generated. Users are then informed based on the classification outcomes.

The mobile application is developed using Flutter for Android (Kuzmin et al., 2020). The output weights files from the model training, saved as .weight and .cfg files, are converted into a .tflite (TensorFlow Lite) file and utilized in the developed mobile application. The .tflite file returns recognition and classification data from the real-time capture in a string expression, which is then parsed and employed in the relevant functions. This approach enables instant classification of captures from the camera and issuance of relevant commands to the user via audio prompts.

(a) (b)



(c) (d) (e)

Figure 4. Mobile application screenshots. (a) Example of red light. (b) Example of green light. (c) Example of crosswalk. (d) Personal info page. (e) Help page

The objective is to design a mobile application that is ergonomically tailored for visually impaired individuals, taking into account their user experience. Upon launching the application, it establishes a connection with the camera and begins processing incoming images. If a red light is detected, a voice guide notifies the user, providing messages like "Red light, please wait" or "Please continue waiting" until the red light changes. When a green light is detected, the voice guide signals the user with "You can pass." Similarly, upon detecting a crosswalk, the voice guide alerts the user with "Crosswalk is detected." Figure 4 displays screenshots of the application.

The text within the mobile application and its user guide are synchronized with the phone's voice description feature, ensuring ease of use even for individuals with complete visual impairment. Accordingly, the font sizes in the mobile application are adjusted to a level suitable for normal vision reading.

4. Results and Discussion

Figure 5 shows the percent accuracy obtained by testing the dataset on the trained model.

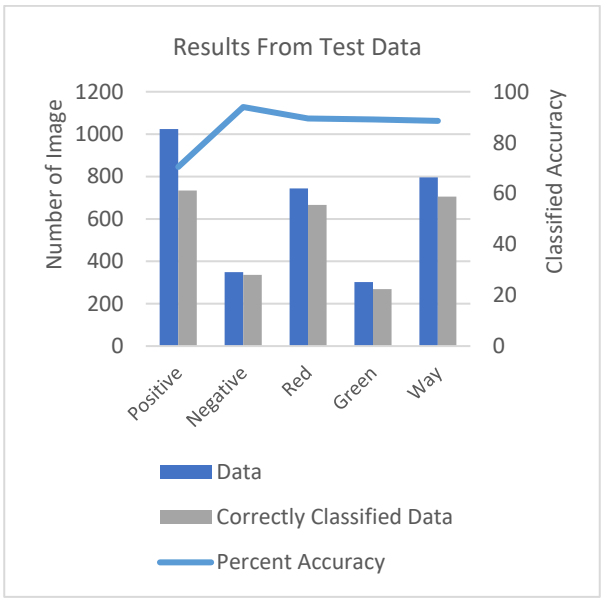


Figure 5. Results from test data

The highest accuracy, at 94%, is achieved in the negative class, while the lowest accuracy, at 70.4%, is observed in the positive class after testing each class. These two classes are ancillary to the primary focus of this study and are solely utilized for distinguishing between pedestrian and vehicle traffic lights. The main parameters of interest in this study-red, green, and crosswalk classes-demonstrate accuracies of 89.52%, 89.1%, and 88.57%, respectively. In Figure 5, the classes are defined as follows: Positive (Detected pedestrian traffic lights), Negative (Detected vehicle traffic lights), Red (Red light), Green (Green Light), and Crosswalk.

Figure 6 illustrates the loss chart of the trained model, indicating that the average loss value decreased to 0.20 by the end of the training process. To enhance accuracy ratios, the dataset can be updated and expanded for re-training the model, or the same dataset can be utilized for further training to achieve a lower average loss value. The utilization of the YOLOv2-Tiny algorithm version resulted in a slightly lower accuracy ratio in this mobile application. However, leveraging the Tiny version facilitated faster response times for real-time image processing. Opting for higher YOLO versions over YOLOv2 Tiny would enhance accuracy. Yet, for this study, YOLOv2 Tiny was chosen due to its compatibility with the Flutter SDK. Future studies may lean towards versions with higher accuracy using different mobile application development platforms.

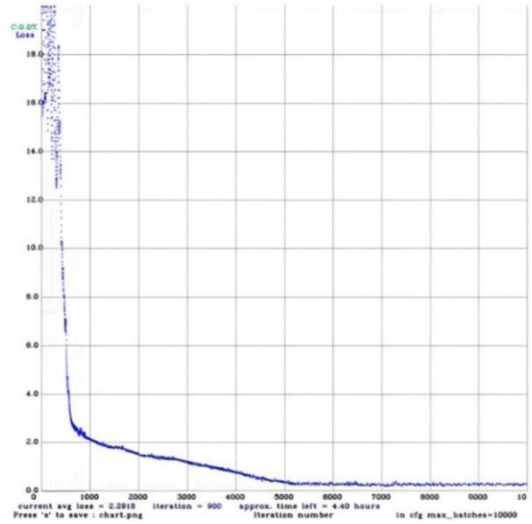


Figure 6. Loss chart

(a) (b)

Figure 7. Test photos of the trained model. (a) Rainy day and long-distance example. (b) Insufficient lighting example. (c) Example of the negative class. (d) Close distance example.

The model's operational accuracy may decrease upon integration into a mobile application. Several factors contribute to this decline. Firstly, the quality of the captured image is crucial; clearer images yield more accurate results. Image clarity hinges on three primary factors: ambient lighting, camera quality, and weather conditions. Images taken in dimly lit environments tend to be less clear compared to those taken in adequately illuminated settings. Conversely, extremely sunny weather can alter image detection due to sunlight reflections. Camera quality directly affects image clarity. In this study, the mobile phone's positioning is also significant; it is recommended to hang or hold the phone around the user's chest, ensuring alignment between the user and the phone's direction. Accuracy might vary in heavy rain due to camera wetness or reduced visibility. Addressing such issues may involve developing filtering

methods tailored to different environmental conditions. Figure 7 displays the model's performance under various scenarios.

Table 2 presents a comparison between this study and previous ones. The comparison encompasses the platform, methodology, classified objects, and results obtained. Notably, unlike prior applications, our mobile application classifies both parameters (Pedestrian Crossing Lights and Zebra Crossing).

The Cross-Assist application stands out from others, as shown in the Table 2, because it is a mobile application with a high accuracy rate and the capability to identify both traffic lights and crosswalks. While models created using the CNN method demonstrate superior accuracy rates, they are typically run on a GPU unit rather than a mobile application, which is less ergonomic in practice.

Table 2. Comparison of studies

Article	Platform	Method	Detected Object	Results
Cross-Assist	Mobile App.	YOLOv2 Tiny	Pedestrian Crossing Lights	89% accuracy
			Zebra Crossing	89% accuracy
Li et al., 2020	Portable GPU Unit	CNN	Pedestrian Crossing Lights	94% accuracy
Shangguan et al., 2014	Mobile App.	Hough Transform	Zebra Crossing	90% accuracy
Cheng et al., 2018	Stable camera on glasses	SVM	Pedestrian Crossing Lights	74% recall, 98% precision
Cheng et al., 2017	Glasses	CNN	Zebra Crossing	60% recall, 85% precision
Yu et al., 2019	Wearable device	CNN	Pedestrian Crossing Lights	94% accuracy
Moura et al., 2022	N/A	CNN	Pedestrian Crossing Lights and Zebra Crossing	95% accuracy

The Cross-Assist application is also tested in a real world scenario with images taken from different distances on a 4-lane road with a 12 MP camera, both day and night. Table 3 shows the mean accuracies obtained from 14 meter, 7 meter and 3.5 meter away from the target traffic light and crosswalk. It can be seen from the table that results obtained during day time are higher than the night time. When the target to be classified is close the application gives much higher recognition accuracies. This is because when the target to be recognized is far away, the camera perceives a wide perspective that includes many different contents. However, as the person approaches the traffic light and crosswalk, the application's target recognition accuracy increases to 90%, proving the reliability and accuracy of the application.

Table 3. Test results obtained from real-world

Class	Day			Night		
	14 m	7 m	3.5 m	14 m	7 m	3.5 m
Positive	%55	%75	%90	%50	%65	%85
Negative	%45	%65	%85	%35	%55	%75
Crosswalk	%45	%70	%85	%35	%60	%75
Red Light	%50	%75	%95	%40	%65	%90
Green Light	%45	%75	%90	%35	%65	%90

5. Conclusion

This study aims to create a mobile assistant to ease the lives of visually impaired people by a developed mobile application that provides real-time voice alerts about the status of crosswalks and traffic lights. The model in this study is trained with the YOLOv2-Tiny algorithm, known for its success in real-time image processing. The trained model is tried out with test data. In these trials, it was observed that the model works with 90% success accuracy. Training the model with a more extensive dataset, using higher YOLO versions, and thus using different mobile application development platforms would increase the accuracy. In this study, the mobile application is developed with Flutter. It was ergonomically designed considering the physical condition of the user.

A limitation of this study is that its operational accuracy is influenced by factors such as image quality, lighting, weather conditions, and the positioning of the mobile phone. Sharper images lead to more precise results, while images captured in poorly lit environments or under very bright sunlight can reduce success rates. In heavy rain, accuracy may be affected by camera wetness or diminished visibility. Additionally, camera quality and the placement of the mobile phone play a crucial role in performance.

With this mobile application, a visually impaired person will be protected from the danger of traffic, and their participation in social life will be made easier. This study, which began with the idea of "barrier-free living" as the

goal of today's modern world, brings that goal closer. In future studies, the user's address and directions can be added to the application using the phone's GPS feature.

6. Acknowledgments

This research is derived from the Bachelor's Thesis of the Department of Electrical and Electronics Engineering at Faculty of Technology, Gazi University.

Conflict of Interest

The authors declared that there is no conflict of interest.

Author Contributions

The authors declare that they have contributed equally to the article.

7. References

Arora, A., Grover, A., Chugh, R., & Reka, S. S. (2019). Real-time multi-object detection for the blind using single shot multibox detector. *Wireless Personal Communications, 107*(1), 651-661. doi.org/10.1007/s11277-019-06604-5

Ash, R., Ofri, D., Brokman, J., Friedman, I., & Moshe, Y. (2018). Real-time pedestrian traffic light detection. *2018 IEEE International Conference on the Science of Electrical Engineering in Israel (ICSEE)*. December 12-14, Eilat, Israel, 1-5. doi.org/10.1109/ICSEE.2018.8553696

Aydın, S., Samet, R., & Bay, Ö. F. (2020). A survey on parallel image processing studies using CUDA platform in GPU programming. *Journal of Polytechnic*, *23*(3), 737-754. doi.org/10.2339/politeknik.576835

Chen, C., Min, H., Peng, Y., Yang, Y., & Wang, Z. (2022). An intelligent real-time object detection system on drones. *Applied Sciences*, *12*(20), 10227.

Cheng, C.-C., & Tsai, C.-C. (2024). A visually assistive guidance system for visually impaired pedestrians passing crosswalks. *2024 International Conference of Control Systems, and Robotics (CDSR 2024)*. June 10-12, Toronto, Canada, Paper No. 112. doi.org/10.11159/cdsr24.112

Cheng, R., Wang, K., Yang, K., Long, N., & Hu, W. (2017). Crosswalk navigation for people with visual impairments on a wearable device. *Journal of Electronic Imaging, 26*(5), 053025. doi.org/10.1117/1.JEI.26.5.053025

Cheng, R., Wang, K., Yang, K., Long, N., & Liu, D. (2018). Real-time pedestrian crossing lights detection algorithm for the visually impaired. *Multimedia Tools and Applications, 77*(16), 20651-20671. doi.org/10.1007/s11042-018-6181-8

Dionisi, A., Sardini, E., & Serpelloni, M. (2012). Wearable object detection system for the blind. *2012 IEEE International Instrumentation and Measurement Technology Conference Proceedings*. May 13-16, Graz, Austria, 1255-1258.

Francies, M. L., Mohamed, M. A., & Mohamed, A. M. (2022). A robust multiclass 3D object recognition based on modern YOLO deep learning algorithms. *Concurrency and Computation: Practice and Experience*, e6517. doi.org/10.1002/cpe.6517

Ghilardi, M. C., Simoes, G., Wehrmann, J., Manssour, I. H., & Barros, R. C. (2018). Real-time detection of pedestrian traffic lights for visually-impaired people. *2018 International Joint Conference on Neural Networks (IJCNN)*. Jul 08-13, Rio de Janeiro, Brazil, 1-8. doi.org/10.1109/IJCNN.2018.8489628

Huang, R., Pedoeem, J., & Chen, C. (2018). Yolo-Lite: A real-time object detection algorithm optimized for non-GPU computers. *2018 IEEE International Conference on Big Data*. December 10-13, Seattle, WA, USA, 2503-2510. doi.org/10.1109/BigData.2018.8622624

Hwang, H., Kwon, S., Kim, Y., & Kim, D. (2024) Is it safe to cross? Interpretable Risk Assessment with GPT-4V for Safety-Aware Street Crossing. *arXiv. 2402.06794v2*. doi.org/10.48550/arXiv.2402.06794

Kaggle (2022). Traffic Light Detection Dataset. Retrieved Jun 07, 2024, from https://www.kaggle.com/datasets/wjybuqi/traffic-light-detection-dataset.

Kaggle (2020). Crosswalk-Dataset. Retrieved Jun 07, 2024, from https://www.kaggle.com/datasets/davidsilvam/crosswalkdataset.

Khan, S., Rahmani, H., Shah, S. A. A., & Bennamoun, M. (2018). A guide to convolutional neural networks for computer vision. *Synthesis Lectures on Computer Vision, 8*(1), 1-207. doi.org/10.2200/S00839ED1V01Y201801COV014

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*. doi.org/10.1016/j.eswa.2022.118720.

Kuzmin, N., Ignatiev, K., & Grafov, D. (2020). Experience of developing a mobile application using Flutter. In *Information Science and Applications.* (pp. 571-575). doi.org/10.1007/978-981-15-6204-3_58

Li, X., Cui, H., Rizzo, J.-R., Wong, E., & Fang, Y. (2020). Cross-safe: A computer vision-based approach to make all intersection-related pedestrian signals accessible for the visually impaired. In *Advances in Intelligent Systems and Computing.* (pp. 132–146). doi.org/10.1007/978-3-030-40549-3_15

Li, Y., Li, J., & Meng, P. (2023). Attention-YOLOV4: A real-time and high-accurate traffic sign detection algorithm. *Multimedia Tools and Applications, 82*, 7567-7582. doi.org/10.1007/s11042-022-12150-2

Mahesh, T. Y., Parvathy, S. S., Thomas, S., Thomas, S. R., & Sebastian, T. (2021). Cicerone-A Real Time Object Detection for Visually Impaired People. *IOP Conference Series: Materials Science and Engineering, 1085*(1), 012006. doi.org/10.1088/1757-899X/1085/1/012006

Moura, R. S., Sanches, S. R. R., Bugatti, P. H., & Saito, P. T. (2022). Pedestrian traffic lights and crosswalk identification. *Multimedia Tools and Applications, 81,* 16497-16513. doi.org/10.1007/s11042-021-11817-8

Rajwani, R., Purswani, D., Kalinani, P., Ramchandani, D., & Dokare, I. (2018). Proposed system on object detection for visually impaired people. *International Journal of Information Technology*, *4*(1), 1-6.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *IEEE Conference On Computer Vision and Pattern Recognition*, Jun 26-July 1, Las Vegas, USA, 779-788.

Shangguan, L., Yang, Z., Zhou, Z., Zheng, X., Wu, C., & Liu, Y. (2014). Crossnavi: Enabling real-time crossroad navigation for the blind with commodity phones. *2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, Sep 13-17, Seattle, WA, USA, 787-798.

Sinha, R. K., Pandey, R., & Pattnaik, R. (2018). Deep learning for computer vision tasks: A review. *arXiv preprint* arXiv:1804.03928.

Son, H., & Weiland, J. (2022). Wearable system to guide crosswalk navigation for people with visual impairment. *Frontiers in Electronics, 2,* 790081. doi.org/10.3389/felct.2021.790081

Srinivas, S., Sarvadevabhatla, R. K., Mopuri, K. R., Prabhu, N., Kruthiventi, S. S. S., & Babu, R. V. (2016). A taxonomy of deep convolutional neural nets for computer vision. *Frontiers in Robotics and AI, 2,* 36. doi.org/10.3389/frobt.2015.00036

Tosun, S., & Karaarslan, E. (2018). Real-time object detection application for visually impaired people: Third eye. *2018 International Conference on Artificial Intelligence and Data Processing (IDAP)*, Sep 28-30, Malatya, Turkey, 1-6.

Wang, Y., Mao, K., Chen, T., Yin, Y., Chen, G., & He, S. (2021). Accelerating real-time object detection in high-resolution video surveillance. *Concurrency and Computation: Practice and Experience*, e6307. doi.org/10.1002/cpe.6307.

Wang, J., Wang, S., & Zhang, Y. (2023) Artificial intelligence for visually impaired. *Displays, 77*, 102391, doi.org/10.1016/j.displa.2023.102391.

Yu, S., Lee, H., & Kim, J. (2019). Lytnet: A convolutional neural network for real-time pedestrian traffic lights and zebra crossing recognition for the visually impaired. *International Conference on Computer Analysis of Images and Patterns*, Sep 28-30, Salerno, Italy, 259-270.