A Smartphone Based Real-Time Object Recognition System for Visually Impaired People

Md. Atikur Rahman, Kazi Md. Rokibul Alam(✉), and Muhammad Sheikh Sadi

Khulna University of Engineering and Technology, Khulna, Bangladesh
rahman1907511@stud.kuet.ac.bd, {rokib,sadi}@cse.kuet.ac.bd

Abstract. Visual impairment is a major crisis for visually impaired people (VIP) and consequently, VIP require keen guidance to perform their regular activities. This paper develops a smartphone-based object (*e.g.* currency note, staircase, etc.) recognition system that can alleviate the monetary transactions, mobility issues, etc. for VIP. Due to the usage of a singular smartphone, the design of the system remains straightforward, and it requires no extra hardware. Alongside, it is convenient to adapt to the human body. Then, to recognize the object in real-time, it exploits Single Shot Detector (SSD), Convolutional Neural Network (CNN), and Tensorflow-lite (tflite) that classifies, train as well as test the object, and supports the platform, respectively for this purpose. First, it creates a dataset in the format of the COCO dataset. Then, it labels the recognized object by a text to speech conversion method and sends it to the VIP via Bluetooth technology. The experimental results show that for object recognition and detection, the accuracy of the developed system is 94.25% and 98.23%, respectively that outperforming the existing systems. Moreover, it uploads all processed data to a remote server. Overall, the proposed system uses audio messages to guide the VIP in recognizing currency notes and avoiding obstacles in their surroundings.

Keywords: Visually Impaired · Computer Vision · Object Detection · Object Recognition · Tensorflow-Lite

1 Introduction

Vision is one of the fundamental human senses, and it assumes a huge part in human sensation about the general circumstance. For visually impeded people (VIP) to have the option to give, experience their vision, creative mind portability is necessary. The International Classification of Diseases 11 (2018) characterizes vision debilitation into two classes, distance and close introducing vision hindrance [1]. Globally, the main sources of vision hindrance are cataracts, glaucoma, age-related macular degeneration, uncorrected refractive error, corneal opaqueness, diabetic retinopathy, trachoma, eye wounds [2], etc. It restricts visually impaired capacity to explore, perform ordinary undertakings, and influence their personal satisfaction and capacity to communicate with the encompassing scene upon independence.

Md. S. Satu et al. (Eds.): MIET 2022, LNICST 490, pp. 524–538, 2023.
https://doi.org/10.1007/978-3-031-34619-4_41

Around 253 million VIP are in the world which is indicated by the statistics of the World Health Organization (WHO). Most of them have a close or distance vision weakness. In something like 1 billion or close to half of these cases, vision disability might have been forestalled or still can't seem to be addressed [3]. The most miserable problem faced by VIP is not knowing the distance of obstacles around them in a new environment when they didn't visit there before [4]. Being unable to follow out and keep away from blockages in their courses, most frequently they become the casualty of a few undesirable inconveniences that could lead them to emotive wretchedness or unasked situations and their incessant versatility is being undermined [5]. They ought to get the object's location in their course in order to ensure danger-free movement. So they need a low-cost, lightweight, and compact assistive device to do their daily activities properly and smoothly.

In daily transactions, the VIP face a great problem to perceive currency notes on account of the similarity of paper surface and size among different groups [6]. Simultaneously, VIP are confronting a serious problem with recently launched currency notes texture and sizes [7]. For instance, the newly released BDT Fifty and BDT two thousand's notes have indistinguishable colors and textures, making it challenging for the VIP to recognize and make proper exchanges. This issue causes an extraordinary experience when they exchange their currency in shops or other fields [8].

Recognizing plates of stairs is a matter of worry for the VIP since the inability to distinguish them can cause serious accidents [9]. Without seeing the step, it is very difficult to impeccably distinguish the edges of each plate of a step [10]. Among different issues looked at by VIP are perceiving washrooms, chairs, tables, etc.

In this paper, we develop a smartphone based object identification system that makes monetary transaction, movement, etc. easier for VIP. We also propose an algorithm to measure the distance between the object and the VIP via a camera. The rest of the paper is arranged as follows. Section 2 studies related works. The methodology is described in Sect. 3. The Android based application that is developed is clarified in Sect. 4. Section 5 shows the experimental analysis, and finally, Sect. 6 concludes the paper.

2 Related Work

A number of researchers targeted to minimize the difficulties faced by the VIP. In the domain of object detection and recognition, some ones used sensor-based systems [1–3], some ones used computer vision-based systems [4–6] and other ones worked with smartphone platforms [7–9]. Among them, some remarkable works are as follows.

Dahiya et al. [10] proposed a deep learning based system that can identify public convenience in Restrooms, stations, and ATMs. It uses Faster R-CNN and resnet50 to recognize the sign of public convenience, can train over 450 images and gains the accuracy of 92.13%. But, it didn't consider the currency notes.

Yadav et al. [11] presents a navigation stick that can detect object and identify obstacles. It consists of a DSP processor, several sensors, and a camera, can recognize objects using machine learning tools, uses sensors to measure distance and detect wet floors. But, its design is bulky and the hardware cannot be mounted on a single board.

Habib et al. [12] proposed a walking stick that consists of a raspberry pi, an R-GBD camera, and an ultrasonic sensor. It developed a hybrid system to recognize staircases using pre-trained images where the accuracy was 98.73%. However, its hardware design was not practical enough due to mounting on a single walking stick.

Salunkhe et al. [13] proposed an android application that can identify the objects in the surroundings. The real object identification is done by the tflite API using the main camera to capture the image and its overall accuracy is around 90%.

Badave et al. [14] developed a smartphone-based system that consists of a camera, an audio device, and an android application. The system sends the name of the recognized object with its distance to the user using headphones. Its accuracy is 87% and also didn't consider the currency notes.

3 Proposed Methodology

The principle approach of the proposed system is partitioned into three separate layers. The first layer is the Object Detection. An algorithm is applied herein to detect the object and measure the distance. The Object Recognition module is the second layer where the recognition procedure discusses from dataset creation to image classification. Here, You Only Look Once (YOLO) neural network, SSD with tflite is used in the recognition module to gain the highest processing speed and accuracy. The third layer is the Cloud layer. The measured distance and label of the recognized object are sent to an online server for storage and analysis purposes.

3.1 Object Detection Layer

In the proposed system, a single camera is used for object detection and to measure the distance of the object from the VIP. Now, camera technology has two different types: Charged Coupled Device (CCD) [15] and Complementary Metal Oxide Semiconductor (CMOS) [16]. Currently, the most fundamental type utilized in mechanical vision systems is CCD camera technology. The perusing process is performed at one corner of the CCD chip. This implies that each charge ought to be economically moved across the chip in succession and a section to arrive at one explicit corner. This strategy requires an exact procedure to guarantee the steadiness of the shipped charge. So this technology is used in those sectors where high computational power is available.

The CMOS camera technology has a similar array of pixels as CCD cameras, however with a few semiconductors alongside every pixel. During the information perusing process in CMOS cameras, all pixels magnify each pixel's sign in the array. This interaction goes on until the target is achieved and there is a compelling reason need to move every pixel's charge down to the particular area. The CMOS technology has a simpler setup than CCD technology which empowers CMOS cameras to consume fundamentally less power. This lower power utilization makes CMOS innovation reasonable for use in the field of embedded systems and smart-phone.

In this paper, we use the CMOS technology-based camera to capture the real-time object. An algorithm is used to measure the distance. Figure 1 shows the flowchart that represents the algorithm to measure the distance using a single camera. The overall procedure of distance measurement is discussed below.

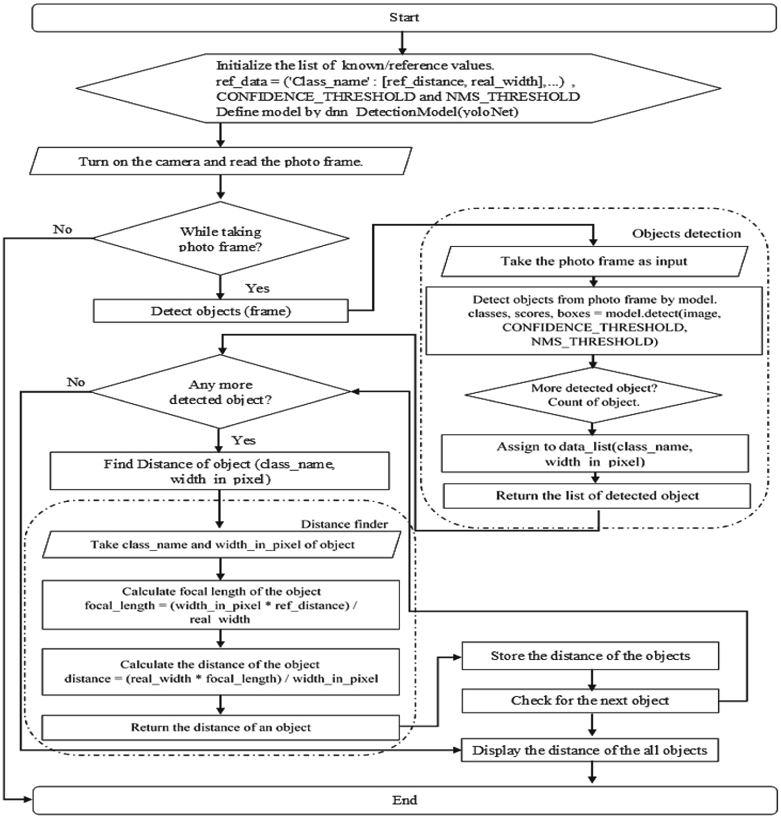
Start
Initialize the list of known/reference values.
ref_data = ('Class_name' : [ref_distance, real_width],...) ,
CONFIDENCE_THRESHOLD and NMS_THRESHOLD
Define model by dnn DetectionModel(yoloNet)
Turn on the camera and read the photo frame.
No
While taking photo frame?
Yes
Detect objects (frame)
Objects detection
Take the photo frame as input
Detect objects from photo frame by model.
classes, scores, boxes = model.detect(image, CONFIDENCE_THRESHOLD, NMS_THRESHOLD)
More detected object? Count of object.
Assign to data_list(class_name, width in pixel)
Return the list of detected object
No
Any more detected object?
Yes
Find Distance of object (class_name, width in pixel)
Distance finder
Take class_name and width_in_pixel of object
Calculate focal length of the object
focal_length = (width_in_pixel * ref_distance) / real width
Calculate the distance of the object
distance = (real_width * focal_length) / width_in_pixel
Return the distance of an object
Store the distance of the objects
Check for the next object
Display the distance of the all objects
End

Fig. 1. Flowchart of distance measurement module using camera.

Step 1. At the beginning of the distance measurement, we set the height of the VIP from the ground. Actually, this height will be the place where the smartphone is attached to the VIP. This height varies from VIP to VIP. Then we choose several reference objects and set the actual height, width, and known distance from the VIP of those objects.

Step 2. In this step, we select the camera for the distance measurement. In this step, we choose the rear side camera module of the smartphone and fix the main camera sensor. If the smartphone contains several camera sensors, this will automatically select the main sensor for distance measurement purposes.

Step 3. The CMOS sensor collects the real-time instances from the environment. An object detection module is required to recognize the object from the instances. We use the same recognition procedure which will be discussed in the "Object Recognition Layer" to recognize the objects in real-time. After recognition, this step returns the measured height and width of the object using a bounding box.

Step 4. In this step, we measure the focal length using the known distance, actual width, and the measured width of the object. The value of the focal length will be passed to the next step to measure the actual distance of the object from the VIP. The calculation of focal length is given in Eq. 1. The first width and distance of Eq. 1 is the reference object distance which was set in step 1.

$$focal\_length = (width_{ref\_Image} * distance_{ref\_Image})/width_{actual} \tag{1}$$

Step 5. This is the final step of distance measurement. In the step, we use a distance finding method which is measured by using Eq. 2.

$$distance = (width_{actual} * focal\_length)/width_{measured} \tag{2}$$

In Eq. 2, we set the actual width at step 1, the focal length is calculated at step 4, and the width of the object is measured at step 3 from the collected image frame by the CMOS sensor.

3.2 Object Recognition Layer

The object recognition module is capable of recognizing more than hundred different types of objects in a real-time scenario. We have used MS COCO dataset for training and testing purposes. The MS COCO dataset [17] is an object identification, captioning, and segmentation dataset distributed by Microsoft. AI and Computer Vision designs prevalently utilize the COCO dataset for different machine learning projects. Moreover, the COCO dataset provides several properties-eighty object categories, five captions for each image, super pixel object segmentation, etc. In this paper, we focus on Bangladesh currency notes and staircase detection which are not available in the COCO dataset. So we create our own customize dataset same as the COCO dataset format. The overall process from dataset creation to object recognition is discussed briefly below.

Dataset Creation

The dataset creation process is partitioned into four steps. The first step is to collect the image of the referenced object. Image augmentation is the second step. An application is used to annotate those images which are discussed in the third step. Among the following steps, step 4 depicts the conversion of the dataset.

Step 1. The images of the currency notes currently running in Bangladesh are collected from the corporate branch of the Janata Bank. We have followed several conditions to make the better performance of the dataset. Namely, we have taken at least five different ages notes for each currency, images are taken both in high and low light environments and in hand or on table conditions.

On the other hand, the images of the staircase are collected from the campus (academic, administrative, and dormitory buildings) of Khulna University of Engineering & Technology (KUET), Khulna. All the currency and staircase images are in identical size which is 640 × 480 same as the COCO dataset. The preprocessing of the collected images is performed in this step. We have removed the blurred, high contrasted images from the collection. Those images which are full black due to the absence of light and full white because of sunlight are also removed from the collection.

Step 2. Image augmentation is the process of generating a different image from a raw image using some technique. In this step, we use four augmentation techniques (rotating, vertical flipping, horizontal flipping, and mirroring) to make a huge number of image collections in the dataset. After the augmentation process, we divide the dataset into two parts − one is for training and another part is the testing.

Step 3. We use "LabelImg" tool [15] to annotate every single image of the collection. In the "LabelImg" tool, we mark the required portion of the image using a rectangle (bounding box) and provide a label for that marking object. At the end of this phase, an XML document is generated by the application for each image that provides the object label, resolution, and coordinate of the bounding box (x-min, y-min, x-max, y-max).
Step 4. After the annotation, we get a dataset in PASCAL VOC format from those XML files and a class file that contains all object names. Then we convert the XML format into COCO Jason format using a python program. The Jason file contains all object's information in a single file.

Model Creation
In this phase, at first, we generate a YOLO model that is saved as the weight file in a local directory. Then the YOLO model is converted into TensorFlow (TF) model. The TF is not applicable in Android applications. So we convert the TF model into tflite model at the last step of the model creation phase. The overall process is as below.

Step 1. As already mentioned, YOLO is a real-time object recognition system. In a smartphone-based system YOLO processes real-time instances at 30 FPS on the COCO dataset. In this step, we train the YOLO model on our custom dataset that is generated in the previous phase. During the training period, the weights are saved after 500 successful iterations. After the end of the training, the last saved weights are used as the final weights of the YOLO model.
Step 2. The trained YOLO model is converted into the TF model in this step. A frozen_darknet (.pb) file is generated after the conversion. The internal structure, graph definition, and weights of the model are frozen into that file. The TF model requires high computational power and it gains a better accuracy while object recognition.
Step 3. The TF model is converted into tflite model due to its bulky size and huge power consumption rate. The tflite is generally designed for embedded systems and mobile platforms. After the conversion, the testing segment of the dataset is used to test the final tflite model. We use this tflite model as a pre-trained model for this application.

Classification
The object is recognized in this phase. The system takes real-time image instance from the environment and the recognition module classify the object using the pre-trained model. The overall recognition process is depicted below.

Step 1. In this step, the system uses the rear camera of the smartphone to capture live images. We set the resolution of the input image as 640 × 480. Then the features of the input images are extracted by MobileNet [18] and passed to the detector.
Step 2. The proposed system uses SSD [19] as a detector to recognize objects from input images whose features are extracted in the previous step. The SSD utilizes convolutional layers of different sizes and defaults bounding boxes. The recognition module runs at 59 FPS on the smartphone platform.

3.3 Cloud Layer

The cloud layer enables the system connectivity with a remote server. The processed data (distance of an object from the VIP, recognized label of an object, and free-fall location and time) of the above three layers are stored first in the SQLite database of the smartphone. It is possible to erase all data from the SQLite database if a user resets the phone. Due to the erasable issue, we handle a remote server through the cloud layer. We use the "Volley" HTTP library to establish the connection to the server.

Volley library uses JSON format to send and fetch data. So, we make a JSON format from the raw data. In the proposed application, a POST method is used to send and fetch processed data. The application handles the HTTP request with the POST method to transfer data from the VIP to the server and vice versa.

In the server end, we register a domain and host the corresponding web pages to fetch and send the JSON format data to the user end application. A database is created on the server to store the data. We use integer data type for the object detection layer and String data type for the second layer. The picture of the object recognition layer is stored in "blob" format.

We use a separate data table to store the VIP information along with the VIP's guardian information. At least three guardians' information must be included in the database for each VIP. We also fetch all data from this database to improve the performance of the developed application and check accuracy of the above two layers.

4 Application Development

The proposed system is developed on the Android platform. We use "Android Studio" software to develop it which is based on the JAVA programming language and the design is done by Extensible Markup Language (XML). The minimum Software Development Kit (SDK) version of the application is "Android 5.1". So that almost 98% of android users can install and run the application on their smartphone. Moreover, the people of the developing country cannot afford the cost of a smartphone that runs on the latest SDK version.

At first, several dependencies are added in the Gradle file to activate the functionalities for the VIP such as the volley library used to communicate between the VIP and the server. We use proper permission to access internet connectivity, local storage, camera, etc. in the permission section of the manifest file. The permissions are set to support the privacy of the user. The VIP have full control to restrict the sensor's data, contact information, system state, etc. The application fetches data from the camera, and server after the VIP's authorization.

In the front end of the application, we design two layouts. The first one is for inputting the VIP information such as the height, contact number of the guardians, allowing required permissions, etc. The second layout is mainly designed for the use of the camera module. The object detection and recognition layer use this layout to measure the distance of the object from the VIP and recognize the object label using the camera. In the back end of the application, we set the pre-trained model in the asset folder. The model is trained on a personal computer. Due to the low image processing capability, we attach an external graphics card and a solid-state drive to the computer.

We use Bluetooth technology to establish communication between the headphone and the application. The alert signal, the label of the recognized object, and the distance of the object are sent to the VIP via the Bluetooth media. The developed application is run on a smartphone and the performance is discussed in the following section.

5 Experimental Results and Analysis

The performance of the object recognition and detection layer is performed in both indoor and outdoor environments. We create several areas to take image frames of persons, stairs, and currency notes to measure the performance of the recognition module. The same areas are also used for object detection by a single smartphone camera. The experimental results of all the above layers are discussed one by one. We use the following equations to calculate the accuracy, precision, recall, and f1-score for both the object detection and recognition layer.

$$accuracy = (TP + TN)/(TP + TN + FP + FN) \tag{3}$$

$$precision = TP/(TP + FP) \tag{4}$$

$$recall = TP/(TP + FN) \tag{5}$$

$$f1_{score} = 2 * (recall * precision)/(recall + precision) \tag{6}$$

5.1 Object Detection

In this paper, we use the transfer learning method to recognize objects in different scenarios. To train the recognition module, we create a custom dataset. The dataset provides information on common objects along with currency notes and stairs. The training process is done in a personal computer whose hardware is customized before to cope with the image processing. After the end of the successful training, the trained model is transferred to the asset folder of the application. Then the proposed application is developed with the pre-trained model and the testing is done in several conditions, *i.e.* daylight, low light, and indoor and outdoor environments.

We choose four currency notes whose are difficult to understand by normal people without looking clearly, a staircase, and one common object, *i.e.* person to evaluate the performance of the recognition module. In every scenario, we collect 100 samples to calculate the accuracy, error rate, recall, precision, and F1 score. The confusion matrix of those chosen objects with 100 samples is depicted in Table 1.

In Table 1, it is seen that the proposed system performs better in-person recognition in both daylight and low light condition. In the daylight scenario, the system recognized almost 95 samples successfully whereas it can recognize an average of 91 samples out of 100 samples. We can also see that the system performs the lowest recognition for the 500 Taka currency note in both conditions.

Table 1. Confusion matrix at indoor environment.

Objects	Samples	Day light				Low light			
		TP	TN	FP	FN	TP	TN	FP	FN
Person	100	86	9	1	4	83	8	2	7
Stair	100	85	9	1	5	84	9	1	4
50 ৳	100	85	10	0	5	82	10	0	8
200 ৳	100	84	8	2	6	83	9	1	7
500 ৳	100	83	9	1	7	81	9	1	9
1000 ৳	100	84	8	2	6	82	9	1	8

Table 2. Confusion matrix in the outdoor environment.

Objects	Samples	Daylight				Low light			
		TP	TN	FP	FN	TP	TN	FP	FN
Person	100	88	10	0	2	86	9	1	4
Stair	100	86	10	0	4	84	9	1	6
50 ৳	100	85	9	1	5	85	8	2	5
200 ৳	100	86	8	2	4	85	8	2	5
500 ৳	100	87	9	1	3	86	9	1	4
1000 ৳	100	85	8	2	5	84	8	2	6

We also select the same condition in outdoor environments. But the low light condition is not easy to make without taking images after evening or cloudy weather. So we choose the cloudy weather for taking the real-life experience to evaluate the performance. The confusion matrix of the outdoor environment is illustrated in Table 2. The most recognizing object in Table 2 is same as Table 1. But we can see that the system recognizes the 500 Taka currency note better than indoor environments and it is the second-highest recognition object in Table 2.

In Tables 1 and 2, we divide the total samples into two portions. In the first portion, we take 90 samples with a person object and the other portion consists of 10 samples without a person or totally blank. The proposed system can successfully recognize 86 samples of person out of 90 and 9 samples without person out of 10. The accuracy gained by recognizing the first object "person" in the indoor environment is 95% and 91% in daylight and low light condition, respectively. However, in the outdoor environments, for all parameters' the performance has improved. The accuracy for the same object and same conditions in outdoor environments is 98% and 95%, respectively.

Overall, the object recognition performance in indoor and outdoor environments with daylight and the low light condition is illustrated in Table 3. We choose five parameters (accuracy, error rate, precision, recall, and F1-score) to measure the performance of the selected objects. We can see that the average accuracy of the proposed system is

94.25% and 92.67% in daylight and low light condition, respectively. The performance of individual objects is discussed below.

Table 3. Object recognition performance in the indoor and outdoor environment.

Objects	Environment	Day light					Low light				
		Accuracy	Error rate	Precision	Recall	F1-score	Accuracy	Error rate	Precision	Recall	F1-score
Person	Indoor	95.00	5.00	98.85	95.56	97.18	91.00	9.00	97.65	92.22	94.86
	Outdoor	98.00	2.00	100.00	97.78	98.81	95.00	5.00	98.85	95.56	97.18
Stair	Indoor	94.00	6.00	98.84	94.44	96.59	93.00	7.00	98.82	93.33	96.00
	Outdoor	96.00	4.00	100.00	95.56	97.73	93.00	7.00	98.82	93.33	96.00
50 ৳	Indoor	95.00	5.00	100.00	94.44	97.14	93.00	7.00	98.82	93.33	96.00
	Outdoor	94.00	6.00	98.84	94.44	96.59	93.00	7.00	97.70	94.44	96.05
200 ৳	Indoor	92.00	8.00	97.67	93.33	95.45	92.00	8.00	98.81	92.22	95.40
	Outdoor	94.00	6.00	97.73	95.56	96.63	93.00	7.00	97.70	94.44	96.05
500 ৳	Indoor	92.00	8.00	98.81	92.22	94.40	91.00	9.00	98.78	90.00	94.19
	Outdoor	96.00	4.00	98.86	96.67	97.75	95.00	5.00	98.85	95.56	97.18
1000 ৳	Indoor	92.00	8.00	97.67	93.33	95.45	91.00	9.00	98.80	91.11	94.80
	Outdoor	93.00	7.00	97.70	94.44	96.05	92.00	8.00	97.67	93.33	95.45
Average		94.25	5.75	98.75	94.81	96.65	92.67	7.33	98.44	93.24	95.76

The accuracy of staircase recognition in indoor daylight and the low light condition is 94% and 93%, respectively. The accuracy of daylight conditions has increased in the outdoor scenario which is 96%. But it is seen that the system gains the same accuracy in low light conditions for both environments.

Among the four currency notes, the proposed system performs better for recognizing 500 Taka notes in the indoor environment both day and low light conditions. In the recognition of 50 Taka notes, the system has successfully recognized almost 95 samples out of 100 samples. On the other hand, in the outdoor environment, the recognition percentage of 50 Taka note is 94% whereas the 500 Taka note is 96% and 95% in day and low light, respectively. The accuracy of recognizing the 1000 Taka note is the lowest value among the four currency notes. The texture and the color of 1000 Taka notes are mixed up with the other currency. For example, the color of the old 1000 Taka note is the same as the old 500 Taka note. On average, the 500 Taka note reaches high accuracy among four currency notes.

In Table 4, we can see that all systems use Machine Learning technology but run on different platforms. The above articles [11, 12], and [20] represent sensor-based systems and most of them use the raspberry pi as a processing unit. In [11], the authors used a custom dataset for training and testing the system. On the other hand, the other two works, *i.e.* [12] and [20] used the COCO dataset. The highest accuracy is found in [20], which is 100% to detect a "person" object, but the overall accuracy is 85% to 95%. The accuracy of [12] is 98.73% which is better than the rest of the sensor-based system. The above three sensor-based systems perform well, but most of them are bulky in size due to the usage of external sensors. All parts are separated from each other and connected by wires.

The other articles [10, 13, 14, 21] and the proposed system are developed on the Android platform. In the Android platform, all the parts are mounted on a single circuit

Table 4. A comparison of performance with existing systems.

Author	Platform	Technology	Environment	Dataset	Accuracy (%)
[10]	Android	Faster R-CNN, resnet50	Outdoor	COCO	92.13
[11]	Sensor based	YOLO-v3	Outdoor	Custom	94.00
[12]	Sensor-based	Faster-RCNN	N/A	COCO	98.73
[20]	Raspberry pi based	YOLO v3	N/A	COCO	85–95 100
[13]	Android	Tensorflow lite, SSD	Indoor/Outdoor	COCO	90
[14]	Android	Tensorflow	Indoor/Outdoor	N/A	87
[21]	Smartphone	SSD, Tensorflow	N/A	COCO	–
Proposed System	**Android**	**Tensorflow lite, SSD, YOLO-v3**	**Indoor/Outdoor**	**Custom**	94.25

board and no wire are needed to connect separated parts of the system. The maximum weight of these systems is 180–195 gm whereas the average weight of sensor-based systems is more than 300gm. The accuracy of [10] is 92.13% which is higher than other three android platform-based systems. But the proposed system achieves the highest accuracy among the android platform-based systems which is 94.25%. Moreover, the system [10] uses the COCO dataset to train and test purposes. So that the system will be able to recognize almost 91 different objects, but the proposed system is trained with the custom dataset and able to detect more than 100 objects besides the currency notes.

5.2 Object Detection

In this paper, we use a single sensor to measure the distance of real objects. The smartphone is attached to the body of the VIP. At first, we set the smartphone in front of the face using a rubber framework. Then we take measurements of several objects in the region of 50 cm, 100 cm, 150 cm, 200 cm, and 250 cm. Secondly, we set the phone at the upper portion of the hand of the VIP using an armband to do the same measurement. In this experiment, we are able to measure the distance successfully more than a hundred objects. The performance of the above two operations is depicted in Table 5. In both operations, we take 10 samples for each region. Then the average measured distances are shown into Table 5.

In the head-mounted operation (Operation-1), we measure the distance of a person in 5 different regions. The proposed system calculates the distance of the person and sends the measured value to the VIP using headphones. We see that the system performs better in the region of 100 cm and 50 cm. When the distance is more than 130 cm, the accuracy is decreasing due to the object detection process. Sometimes the camera angle changes

Table 5. A comparison of distance measurement performance of Operation-1 and Operation-2.

Actual Distance (cm)	Operation-1				Operation-2			
	Measured Distance (cm)	Accuracy	Standard Deviation	Variance	Measured Distance (cm)	Accuracy	Standard Deviation	Variance
50	49.54	99.08	0.91	0.83	49.21	98.42	1.11	1.23
100	100.13	99.87	0.12	0.02	98.52	98.52	1.30	1.71
150	148.23	98.82	1.56	2.43	148.50	97.69	1.47	2.24
200	197.00	98.50	1.31	1.71	196.57	98.28	1.21	1.46
250	246.85	98.74	1.99	3.94	245.62	98.24	1.50	2.25
Average		**99.00**	**1.18**	**1.78**		**98.23**	**1.32**	**1.78**

the measured distance value although the smartphone places at the same position. The average accuracy of Operation-1 is 99%.

In Operation-2, the smartphone is mounted on the hand of the VIP which is almost 38 cm below than Operation-1. That's why in this operation, sometimes the system is unable to detect hanging objects. The average accuracy of this Operation is 98.23%. The comparison between Operation-1 and Operation-2 in the domain of accuracy is illustrated in Fig. 2. We can see that in all regions, operation-1 performs better than operation-2. The reason behind this is the coverage area. Operation-1 covers a larger area than operation-2. Although, operation-1 performs better, the VIP prefers operation-2 due to the friendly usability.

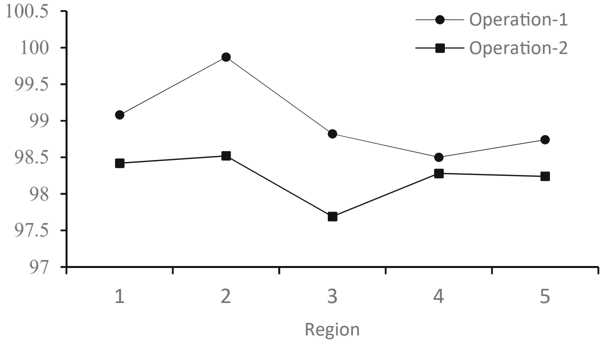

Fig. 2. Comparison of accuracy between operation-1 and operation-2.

The comparison of standard deviation and variance are shown in Fig. 3 and Fig. 4, respectively. In Fig. 3, we can see that operation-1 has the lowest value of standard deviation in region 2, and the highest is in region 5. The variation of standard deviation lies in between 0 to 2. But operation-2 is in a stable location which is 1 to 1.5.

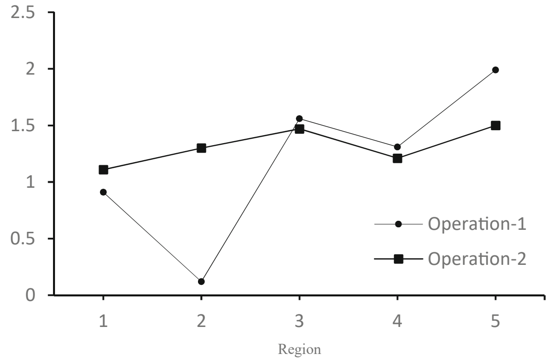

Fig. 3. Comparison of standard deviation between operation-1 and operation-2.

In Fig. 4, the change of the variance is the same pattern of standard deviation. It can be seen that both lines fluctuate in regions 2 and 5. In region 3 the graphical line of operation-1 and operation-2 cross each other. The average variance of both operations are the same which is 1.78.

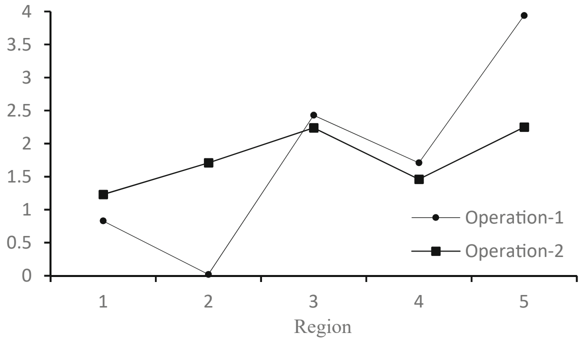

Fig. 4. Comparison of variance between operation-1 and operation-2.

6 Conclusions

This paper develops a smartphone-based object detection and recognition system that can assist VIP in safe movement by identifying objects in their surroundings. The components of the developed system are mounted on a single board and it requires no additional hardware. The weight of the system is reasonable which makes it easy to carry by any aged person. Overall, the accuracy of the proposed system is quite satisfactory and it sends the audio signal to the VIP using wireless technology. Besides, the system can also send a warning notification to guardians of the VIP in case of any unwanted situation. As a whole, the developed system is supportive of the VIP.

Acknowledgement. Information & Communication Technology (ICT) Division of Ministry of Posts, Telecommunications and Information Technology, Bangladesh financially supported Md. Atikur Rahman.

References

1. Farias, G., et al.: A neural network approach for building an obstacle detection model by fusion of proximity sensors data. Sensors **18**(3), 683 (2018)
2. Arslan, O., Koditschek, D.E.: Sensor-based reactive navigation in unknown convex sphere worlds. Int. J. Robot. Res. **38**(2–3), 196–223 (2019)
3. Liu, H., Ma, J., Huang, W.: Sensor-based complete coverage path planning in dynamic environment for cleaning robot. CAAI Trans. Intell. Technol. **3**(1), 65–72 (2018)
4. Fang, R., Cai, C.: Computer vision based obstacle detection and target tracking for autonomous vehicles. In: MATEC Web of Conferences, EDP Sciences (2021)
5. Wang, S.-H., Li, X.-X.: A real-time monocular vision-based obstacle detection. In: 2020 6th International Conference on Control, Automation and Robotics (ICCAR). IEEE (2020)
6. Zhang, Z., et al.: Monocular vision based obstacle avoidance trajectory planning for Unmanned Aerial Vehicle. Aerosp. Sci. Technol. **106**, 106199 (2020)
7. Pavliuk, N., Kharkov, I., Zimuldinov, E., Saprychev, V.: Development of multipurpose mobile platform with a modular structure. In: Ronzhin, A., Shishlakov, V. (eds.) Proceedings of 14th International Conference on Electromechanics and Robotics “Zavalishin’s Readings.” SIST, vol. 154, pp. 137–147. Springer, Singapore (2020). https://doi.org/10.1007/978-981-13-9267-2_12
8. Aksamentov, E., Zakharov, K., Tolopilo, D., Usina, E.: Approach to robotic mobile platform path planning upon analysis of aerial imaging data. In: Ronzhin, A., Shishlakov, V. (eds.) Proceedings of 15th International Conference on Electromechanics and Robotics “Zavalishin’s Readings.” SIST, vol. 187, pp. 93–103. Springer, Singapore (2021). https://doi.org/10.1007/978-981-15-5580-0_7
9. Chen, W., et al.: Novel laser-based obstacle detection for autonomous robots on unstructured terrain. Sensors **20**(18), 5048 (2020)
10. Dahiya, D., Gupta, H., Dutta, M.K.: A deep learning based real time assistive framework for visually impaired. In: 2020 International Conference on Contemporary Computing and Applications (IC3A). IEEE (2020)
11. Yadav, S., et al.: Fusion of object recognition and obstacle detection approach for assisting visually challenged person. In: 2020 43rd International Conference on Telecommunications and Signal Processing (TSP). IEEE (2020)
12. Habib, A., et al.: Staircase detection to guide visually impaired people: a hybrid approach. Revue d’Intelligence Artificielle **33**(5), 327–334 (2019)
13. Salunkhe, A., et al.: Android-based object recognition application for visually impaired. In: ITM Web of Conferences, EDP Sciences (2021)
14. Badave, A., et al.: Android based object detection system for visually impaired. In: 2020 International Conference on Industry 4.0 Technology (I4Tech). IEEE (2020)
15. Siegwart, R., Nourbakhsh, I.R., Scaramuzza, D.: Introduction to Autonomous Mobile Robots. MIT Press (2011)
16. Rigelsford, J.: Introduction to autonomous mobile robots. Ind. Robot: Int. J. (2004)
17. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48

18. Li, Y., et al.: Research on a surface defect detection algorithm based on MobileNet-SSD. Appl. Sci. **8**(9), 1678 (2018)
19. Liu, W., et al.: SSD: single shot multibox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2
20. Shaikh, S., Karale, V., Tawde, G.: Assistive object recognition system for visually impaired. Int. J. Eng. Res. **9**(09), 736–740 (2020)
21. Jakhete, S.A., et al.: Object recognition app for visually impaired. In: 2019 IEEE Pune Section International Conference (PuneCon). IEEE (2019)