Proc. of the 2nd International Conference on Electrical, Communication and Computer Engineering (ICECCE) 12-13 June 2020, Istanbul, Turkey

In-Door Assistant Mobile Application Using CNN and TensorFlow

Nouran Khaled
Faculty of computer science
Misr International University
Cairo, Egypt
nouran1601182@miuegypt.edu.eg

Shehab Mohsen
Faculty of computer science
Misr International University
Cairo, Egypt
shehab1603611@miuegypt.edu.eg

Kareem Emad El-Din
Faculty of computer science
Misr International University
Cairo, Egypt
kareem1608590@miuegypt.edu.eg

Sherif Akram
Faculty of computer science
Misr International University
Cairo, Egypt
sherif1402164@miuegypt.edu.eg

Haytham Metawie
Faculty of computer science
Misr International University
Cairo, Egypt
haytham.metawie@miuegypt.edu.eg

Ammar Mohamed
Faculty of computer science
Misr International University
Cairo, Egypt
Ammar.Ammar@miuegypt.edu.eg

***Abstract*—Visually impaired people struggle to live without assistance or face any aspect of life alone especially with people that cannot afford extra assistance equipment. Usually impaired people receive assistance by either human or wearable devices. The first one bears the burden on the human, while the second adds financial burdens nevertheless the hassle of identifying an object is not decreased. Smartphones are almost accessible to everyone and equipped with accessibility features including sensors that can be utilized to help both visually impaired and sighted people. Thus, this paper proposes an approach using Convolutional Neural Network (CNN), speech recognition and smartphone camera calibration aiming at facilitating the process of indoor guidance for visually impaired people. A smartphone's camera acts as the user's eyes. A pre-trained CNN model is used for object detection and the distance to objects is calculated to guide the user toward the right directions and to warn them of obstacles. The speech recognition part is used as a communication channel between visually impaired people and the smartphone. Also, the proposed approach supports object personalising that helps to distinguish user's item from other items found in the room. To evaluate the personalized objected detection, a customized dataset is created for two objects. The experimental results indicate that the accuracy is 92% and 87% for both objects respectively. Also, we experiment the detect distance of two objects against their real distances. The results achieve 0.05 and 0.08 error ratio.**

***Index Terms*—Mobile application, visually impaired people, object recognition, smartphone.**

I. Introduction

Nowadays according to the World Health Organization [1], visually impaired people are in outrageous growth due to the leading causes of vision impairment, uncorrected refractive errors, and Cataracts. Globally, it is estimated that approximately 1.3 billion people live with some form of vision impairment as shown below in Fig 1. According to the latest survey provided by the World Health Organization, there are more than 2.2 million people with visual impairment in Egypt, 900,000 of which are totally blind. Many solutions have been devised, however, they're either too high in cost which makes them unavailable and affordable to most of the people, or inefficient products that can't be used in natural environments.

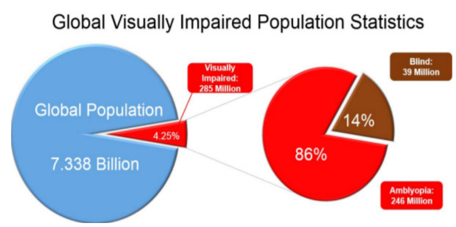

Fig. 1 The ratio of visually impaired people to the world's total population

The biggest challenge for a visually impaired person, especially for someone with complete vision loss according to National Academies Press (US), is to navigate around new places safely and as mentioned by Abbas et al. [2] getting things of independence, which is one of the ultimate goals that a person with impairment might strive to have. Many solutions have been proposed to achieve this goal, but unfortunately most of them are very expensive which makes them out of reach for most of the people, especially in countries with high poverty rates. External assistance might not be available all the time, because people naturally don't like to feel impaired or to be completely dependent on others [3]. Smartphones were considered in this approach due to their high availability nowadays and due to their features and sensors that have high accessibility that could be utilized for the aid of the visually impaired, the camera is utilized to act as the user's eyes.Therefore, this paper will help them a lot in being partially dependent on themselves. It's not safe for visually impaired people to navigate on their own because there might be some obstacles or consequences along their way, as there might be stairs or sharp objects in their surroundings, which might put their life in danger, this approach also tends to the need of object personalization as visually impaired people usually need access to specific items that might not be present in our object detection dataset.

To this end, this paper proposes an approach using the deep learning technique for object detection, text to speech voice recognition and camera calibration.

In this paper, we proposed an indoor assistant [4] using a pre-trained CNN [5], [6] as a type of deep learning methods [7] for object detection to identify the needed object or obstacle, text to speech voice recognition is implemented in order to provide audible interactions with the user for simplicity and ease of use for the visually impaired, we use camera calibration with the object detection model to provide distances to objects and obstacles to accurately provide navigation tips and directions, We use a cloud server to train personalized models for each user using data of their own personal belongings. In the following sections we are going to discuss the related work in Section II, moving on to discussing our proposed methodology explaining the used datasets and the data preprocessing in Section III, then discussing how we tried to achieve this methods in the experimental section in Section IV as well as discussing the achieved results in Section V and proposing our future vision for the project which is found in conclusion and future work Section VI.

II. Related Work

Several research efforts have been proposed to help visually impaired people for example, Milios et al. [8] proposed a mobile application that detects objects for the visually impaired. They used a CNN on the images that the user captures, they then used the predefined model of CNNdroid library to classify the objects from the captured image, the accuracy of Typewriter keyboard was (51.89%) and space bar was (47.91%). The second function that was proposed in this application is banknote recognition, to achieve this goal they chose CraftAR SDK for android. Images of the desired objects are then taken to be recognized and utilized with CraftAR database and SDK. At last, they tested their system on 10 people using a survey, and the results were good in general and the application was well accepted.

Another work was proposed by Liang-Bi et al. [9]. In this wok a wearable glasses is proposed for detecting front objects and a walking stick to guide them through their route, mobile devices application, and on-line information platform to get help in case of collision to the user by sharing their GPS to close friends or family members through mobile device application. Eventually, it is mentioned that integrating deep learning techniques for recognizing images and to develop intelligent walking guiding related functions is on list as future work.

Dhruv et al. [10] used point feature matching to identify common public surroundings. They divided their system into two modules: First one is by creating their template of popular street signs(Pharmacy, special people..etc) and second one is template matching using SURF (Speeded Up Robust Features) which detects signs from captured images and relevant points from the template. Eventually they reached accuracy of 91.67% their problem was variation in color and illumination in captured image or in the template. Their future work includes increasing their accuracy as well as number of common signs and implementing same concept in the indoor navigation.

Meanwhile Laviniu et al. [11] developed a system using smartphone sensors and additional two sensory modules, First one is based on RaspberryPi platform with arm processor and the other one is based on very popular Arduino platform. In outdoors, it uses GPS modules to detect coordinates and move from one point to another then speaks to the user using TTS technology to speak to the user, from the other hand mobile can recognize user commands using google's Voice recognition technology. Inside the building GPS is not available, therefore it uses light sensors present in the phone to do indoor navigation, it also uses accelerometer embedded in the phone to detect the fall of the person and help the call someone to help. Tests made show the efficiency of the system, which can be improved with the development of android based portable devices. Here come our project main goal which is to achieve navigation and assistance for the visually impaired people only using their mobile phone.

Varsha et al. [12] proposed android mobile application to detect objects. First, the user should capture image of the object then enhancements are formed on the captured image like (sharpening..etc). Then, they used CNN which is found in TensorFlow API to detect objects found in the image, the application can work offline as they created their own database which contains 80 classes of objects lastly they used speech synthesis to generate a speech of the detected objects.

Manabu et al. [13] they made a mobile application to detect the most common hazards that could face a visually impaired pedestrian. They trained their dataset to detect stairs, bicycles, crosswalks and sidewalks. If the application detected any of the mentioned items, it should alert the user by vibrating. In order to achieve this they used TensorFlow API and developed their algorithm using CNN, in such experiment they used 637 images including obstacles such as stairs, bicycle, and 393 images of the scenes such as rail tracks, sidewalk and crosswalk were prepared for training. Lastly, they achieved accuracy of 90%.

Finally, Summan et al. [14] proposed an indoor navigation system by applying three main functionalities. First, they used localization as a prerequisite, so that once the user enters a building their current position is fetched and stored in the database. Then, they introduced a user profile where they can store the user's name, gender, height, and most importantly the user's steps length, along with the data from the mobile sensors, such as gyroscope, compass, etc. Finally, path planning is done using the previously mentioned calculated data, along with the building information model, to help guide the user inside the building. The authors tested the system through applying quantitative analysis.

III. METHODOLOGY

The following methods and operations have been carefully chosen to best fit the proposed research, and positively enhance the overall system performance.

A. *Proposed Work*

The proposed approach utilizes the smartphone's camera to recognize the user's face using the facial identification module [15] as an authentication step, also to capture real-time video stream, then perform object detection and distance calculation, the microphone is used to capture the user's audio command using the speech to text module, when the needed object is located, navigation directions are generated and the text to speech module is used to give the user audible directions. If the user is in free roam mode, only alerts are triggered in case the user will meet an obstacle shortly. The approach allows searching for personalized items, that the user can add with the help of a human assistant by capturing a video for this object, and the images are processed and labeled to train a new model in the custom object detection module. A flowchart of the approach is shown in Fig. 2.

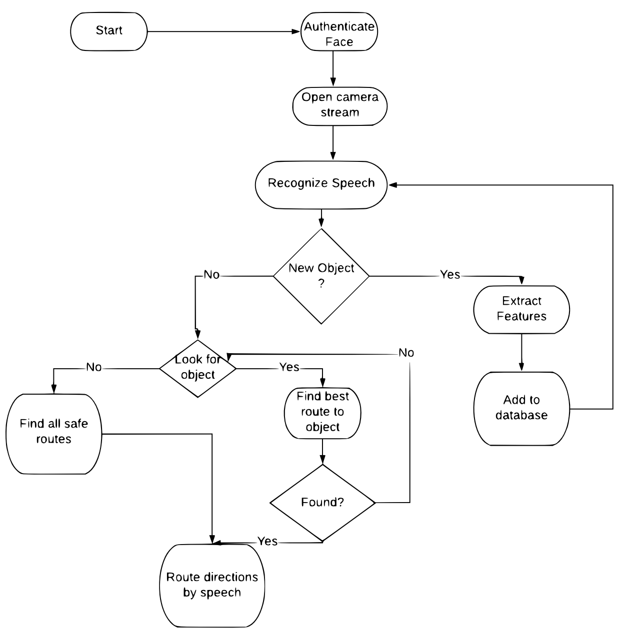

Fig. 2 flowchart of the proposed system

B. *Dataset*

The proposed system is composed of three main datasets listed as follows:

1) **Luxand Dataset**:
Luxand FaceSDK returns the coordinates of all human faces that appear in the picture – or notifies if no face is found. FaceSDK can track all the faces appearing in a video stream. It also allows finding out if a new face appears in the frame, or if one of the subjects leaves the frame. When tested on the FERET dup1+gallery database (frontal passport-like photos), as mentioned in documentation [16]. It successfully detected 99.5% of the faces, with only 0.05% false positives. The proposed system uses this dataset with the face authentication module.

2) **Coco Dataset**:
Common Objects in Context (Coco) dataset, which contains around 123,287 images [17], is a large-scale object dataset made by tensorflow to detect common objects.

3) **Customized Dataset**:
The following dataset items vary according to each user, since that its made of the objects that the user wants to recognize, such as having his very own key that unlocks a specific door. The user's assistant takes 10 images of the desired object from different angles, the system then transforms them with specific width and height coordinates, in order to start the feature extraction process. Finally, the dataset record is generated and linked with the user ID, which was previously generated from the face authentication module.

C. *Data Pre-processing*

The methods used in the commands extraction process are thoroughly explained in this section.

1) **Audio To Speech Conversion**:
We used Google Speech-to-Text which enables developers to convert audio to text by applying powerful neural network models in an easy-to-use API. The API recognizes 120 languages and variants to support your global user base. You can enable voice command-and-control, transcribe audio from call centers, and more. It can process real-time streaming or prerecorded audio, using Google's machine learning technology.

D. *Processing and Classification*

Nowadays, TensorFlow API is one of the most common and accurate APIs in object detection and classification as well as drawing bounding boxes and recognizing 80 different classes [18] of objects as a result it was used in the proposed system.

IV. EXPERIMENTAL SECTION

A. *Classification*

1) ***General Object Detection (Model 1):*** To build a CNN-based model, it is necessary to prepare huge amount of image data for object to be classified to achieve high accuracy, so the proposed general object detection approach uses a pre-trained TensorFlow Lite object classification model for object detection [19], The model is trained on the coco dataset using quantized Mobilenet SSD [20] the dataset used contains only indoor objects such as mobiles, chairs and tables. The model has been trained to detect 80 object categories such as mobiles, people, and chairs. When passed an image, it will output a set of detection results which are discussed in the processing section below, this model crops images to size 300x300 pixels using image transformation with RGB values for each pixel, and returns the location of the detected object, the classification of that object from the available labels and the confidence score that the class was detected successfully and the number of objects in the image, in this approach this model uses video stream from smartphone camera as input.

*2) **Customized Model (Model 2):*** The purpose from this model is to allow users to add their objects to be detected to differentiate between two objects of the same category and to allow users to add objects of new categories. To build a new object dataset the user takes a 30 seconds video which is then divided into 600 frames captured using a smartphone's camera in a well-lit indoor environment. The dataset of each object captured frames is saved as a collection of images. Our training process make use of transfer learning which is the usage of an already trained model to train on your data. This makes the training process taking less time and usually producing better results. For this model we will use Single Shot Detector(SSD) with MobileNet (model optimized for inference on mobile) pre-trained on COCO dataset called SSD MobileNet v2 quantized COCO ,also faster R-CNN is better in the accuracy and more faster, but we cannot use it as cannot be converted to tflite format. To make this TensorFlow model running on a mobile, it is important to start by creating a TensorFlow frozen graph that can be used with TensorFlow lite, then converting the frozen graph to the TensorFlow Lite flatbuffer format which allows the creation of the TensorFlow Lite model which is then assigned to the user after the validation process is a success. The personalized model is then assigned to the user and saved in the database along with the label map so that the user can call the model to detect the object trained whenever needed.

B. ***Navigation***

This proposed module is built upon the two previous models after the user specifies the needed object and it is detected this module takes place.

1) Distance Measurement
   There are three approaches when calculating distance listed as follows:
   a) Using Trigonometric Functions based on Elevation angle
   b) Using Mobile Accelerometer data
   c) Using Triangle Similarity

   The first two approaches depend on the mobile sensors and the angle of which the mobile is hold which results in overhead processing and change of distance value due to minimal movement of hand. The visually impaired person is not guaranteed to hold the phone steadily, so we decided to use the third approach based on Triangle Similarity [21] which depends on simple math and minimal camera calibration [22] to increase process speed. The triangle similarity proves that both real time triangle and the captured triangle between the user and the object are similar and since distance is directly proportional to the pixel width of the object adding the focal length of the camera [23] to the equation helps us deduce the formula for measuring the distance as stated by the equation

$$D = (W \times F)/P$$

   where D is the distance to the object, F is the focal length of the camera, W is the real width of the object saved in a HashMap [24] containing average width of each object and P is the collected width of the object in pixels. The distance is measured in inch and converted into feet to be able to tell the user the number of steps needed to reach object. We calculate the focal length of the smartphone camera used and to get the pixel width of the object we used the width of the recognition bounding box.

2) Obstacle Detection
   To label objects as obstacles we used simple math approach as well after objects are detected and distance is measured to each object the specified object is labeled as needed object while other objects that their distance is less than the threshold which is equal to the distance covered by one human step or if the object area covers more than 80% of the camera's screen area and the object is labeled as an obstacle.

3) Navigation
   The purpose of this module is to safely navigate the user through his/her way to finding their object and it also gives the user an option to safely navigate through the room without a destination. The model also identifies the specified object. If there are multiple objects of the specified type, the user will be navigated to the nearest one. The model will then identify the coordinates of the bounding box for that object and identify its position. And after distance is measured the model will pronounce the direction for the user to take. The model will keep track of the distance, and once it reaches the threshold the user will be notified that the navigation process is finished. If the distance increases between the user and the object, or the object is no longer detected, the user will be notified with a sound message. This model will also navigate the user away from obstacles by pronouncing a direction that contains no objects labeled as obstacles [25].

V. RESULTS & DISCUSSION

A. ***General Object Detection (Model 1) :***

TensorFlow Object Detection API can be used with different pre-trained models. In this work, a SSD model with MobileNet (SSD MobileNet v1 COCO) was chosen. The model had been trained using COCO dataset which consists of 2.5 million labeled instances in 328 000 images, containing 91 object types such as "person" and "bottle" as in Fig. 3. The SSD MobileNet v1 COCO-model is reported to have mean Average Precision (mAP) of 21 on COCO dataset.

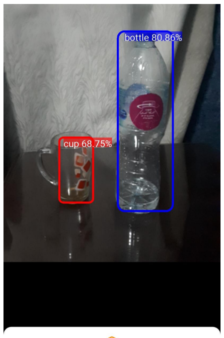

Fig. 3 Object Detection

The pre-trained SSD model was able to recognize the bottle with accuracy of 80% but was less sure about the identity of the cup with accuracy of 68%

B. Customized Model (Model 2) :

1) Prepare Dataset:

The dataset of each object captured frames is saved as a collection of images, these images are big in size, because they have a very high resolution, it was then required to transform [26] them to a lower scale, so that the training process could be faster and fit the input size for SSD quantizied model(300*300). The dataset is split into training set and testing set with ratios of 0.8 and 0.2 respectively. It is vital to create TFRecords that can be served as input data for training of the object detector, so it is needed to extract features of the object from each image by manually tagging the objects in the images using LabelImg as in Fig. 4 and convert it to a TFRecord. Finally, everything is in place and ready to train the model using quantized Mobilenet SSD classifier . Only remaining problem: region proposal methods such as R-CNN are more accurate.

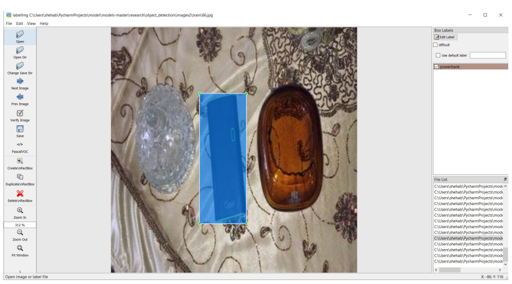

Fig. 4 Trained data image being labeled using labelImg.

1) Training Process:

The pre-trained SSD model (SSD MobileNet v1 COCO) was fine-tuned for our dataset using manually labeled images. A provided configuration file (ssd mobilenet v1 coco.config) was used as a basis for the model configuration (after testing different configuration settings, the default values for configuration parameters were used). The provided checkpoint file for SSD MobileNet v1 COCO was used as a starting point for the finetuning process. The training was stopped after 17400 time-steps when the mPA somewhat leveled out as in Fig. 5. The mAP value increased up to 8000 time-steps. However, the mAP values kept fluctuating even after that, which raised suspicion that even longer training might improve the detection results. Training the model in colab server took over 20 hours with CPU or, alternatively, about 2 hours with GPU for each model. The total loss value was reduced rapidly for models due to starting from the pre-trained checkpoint file as in Fig. 6.

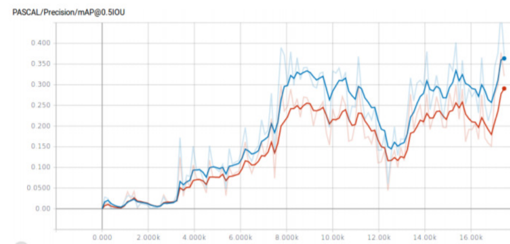

Fig.5 Development of mAP when training model

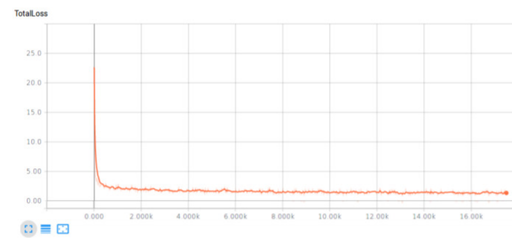

Fig. 6 Development of loss when training model

A concern rose from the analysis of the results that models had not reached the optimal number of training steps at 17400 time-steps (for instance, the fluctuation in mAP level between time-steps as in Fig. 5). Therefore, the models were further trained up to 200,000 time-steps to increase value of mAP.

1) Results:

The shown results in table I are the results of the customized object after training our model.

TABLE I: Accuracy.

Distance Accuracy				
Objects	**Class**	**NImages**	**Input size**	**Accuracy**
ob.1	**Powerbank**	600	300*300	0.92
ob.2	**Book**	600	300*300	0.87

C. Distance Measurement :

Measuring distance results shown in table II using equation1 were satisfying in distances between 50 inches and 100 inches, which was sufficient for guiding the user to objects but insufficient for close range obstacles,hence they were addressed in the navigation module using a threshold for the size of the objects detected on the frame.

TABLE II: Error Ratio.

Distance Accuracy		
Real Distance(inch)	**Detected Distance**	**Error Ratio**
100	95	0.05
50	46	0.08

The proposed system has been thoroughly compared with Summan's et al.'s [14], their system as mentioned earlier in the similar systems section, calculated the distance and fetched the best route to the user with the accuracy mentioned in table III below.

TABLE III: Similar System's Calculated Route Distance.

Paths	User 1	User 2	User 3
	Accuracy	Accuracy	Accuracy
Path 101-105 (42.89m)	0.9	0.8	0.7
Path 104-107 (49.1m)	0.55	0.88	0.5

Through a video stream, the proposed system is able to display the best path available, hence, when compared to the other similar systems, the system provides a higher, more precise accuracy, through a better, more reliable approach shown below in table IV.

TABLE IV: Proposed System's Estimated Distance

Paths	User 1		User 2		User 3	
	Est. Length	Acc.	Est. Length	Acc.	Est. Length	Acc.
Path A (2.54m)	2.4	0.95	2.35	0.92	2.2	0.87
Path B (1.27m)	1.2	0.92	1.1	0.88	1.08	0.85

VI. CONCLUSION & FUTURE WORK

This paper provides a study for building an Android software paper that recognizes indoor objects using a smartphone camera, and returns a navigation guidance to reach the specified object while detecting obstacles, in order to support in-door roaming and locate specified objects processes for visually impaired people.As a future work, implementing this application for iOS developers is also needed so cross platform techniques will be needed [27], we will also improve the performance of the general detection model by increasing the object categories that can be identified. However, if the categories to be detected are increased, the recognition accuracy may be lowered, therefore, some measures for this process are needed and transfer learning [28] methods should be approached. To improve the performance of the obstacle detection model, parallel processing can be used for accelerating the detection process.

VII. ACKNOWLEDGMENTS

The authors would like to finally express their gratitude and would like to thank Engineer Salama Mohamed, because of all the great help and assistance he had provided in guiding us toward tackling the most common issues for the visually impaired people.

REFERENCES

[1] "Vision impairment and blindness," World Health Organization, https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment, 2019.

[2] R. Y. F. B. Abbas Riazi, Fatemeh Riazi, "Outdoor difficulties experienced by a group of visually impaired iranian people," *journal of current ophthalmology*, p. 85–90., 2016.

[3] "Challenges blind people face when living life," 2019.

[4] G. Bohouta and V. Këpuska, "Next-generation of virtual personal assistants (microsoft cortana, apple siri, amazon alexa and google home)," 01 2018.

[5] N. M. D. R. Yamashita, R., "Convolutional neural networks: an overview and application in radiology.," *Insights Imaging 9*, p. 611–629, 06 2018.

[6] A. Samkaria, "Object detection using convolutional neural networks in tensor flow," *INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN TECHNOLOGY*, vol. 5, pp. 351–353, September 2018.

[7] B. Y. . H. G. LeCun, Y., "Deep learning.," *Nature 521*, p. 436–444, 05 2015.

[8] E. Y. E. K. J. E. H. M. M. Milios Awad, Tarek Mahmoud, "Intelligent eye: A mobile application for assisting blind people," *2018 9th IEEE Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON)*, p. 6, June 2018.

[9] M.-C. C. W.-J. C. C.-H. Y. C.-Y. S. Liang-Bi Chen, Jian-Ping Su, "An implementation of an intelligent assistance system for visually impaired/blind people," *2019 IEEE International Conference on Consumer Electronics (ICCE)*, p. 2, 2019.

[10] M. K. D.-K. P. K. Dhruv Dahiya, Ashish Issac, "Computer vision technique for scene captioning to provide assistance to visually impaired," *2018 41st International Conference on Telecommunications and Signal Processing (TSP)*, p. 4, September 2018.

[11] P. S. A. G. Laviniu Ţepelea, Virgil Tiponuţ, "Multicore portable system for assisting visually impaired people," *2014 14th International Workshop on Cellular Nanoscale Networks and their Applications (CNNA)*, p. 2, 2014.

[12] S. J. S. J. Varsha Sharma, Chaitanya Sharma, "Assitance application for visually impaired - vision," *International Journal of Scientific Research and Engineering Development*, p. 4, November 2019.

[13] I. T. C. O. K. K. Manabu Shimakawa, Kosei Matsushita, "Smartphone apps of obstacle detection for visually impaired and its evaluation," *ACIT 2019: Proceedings of the 7th ACIS International Conference on Applied Computing and Information Technology*, p. 6, May 2019.

[14] S. A. F. A. Summan Zaib, Shah Khusro, "Smartphone based indoor navigation for blind persons using user profile and simplified building information model," *2019 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*, p. 6, 2019.

[15] S. Kak, F. Mustafa, and P. Valente, "A review of person recognition based on face model," vol. 4, pp. 157–168, 01 2018.

[16] "Detect and recognize faces and facial features with luxand facesdk," Luxand, Inc., https://www.luxand.com/facesdk/facedetection, 2020.

[17] "Common objects in context," COCO, http://cocodataset.org/explore, 2017.

[18] "Object detection: Tensorflow lite," TensorFlow, https://www.tensorflow.org/lite/models, 2020.

[19] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: A system for large-scale machine learning," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pp. 265–283, 2016.

[20] E. Dong, Y. Lu, and S. Du, "An improved ssd algorithm and its mobile terminal implementation," pp. 2319–2324, 08 2019.

[21] R. P. Prasetya and F. Utaminingrum, "Triangle similarity approach for detecting eyeball movement," *2017 5th International Symposium on Computational and Business Intelligence (ISCBI)*, pp. 37–40, 2017.

[22] W. Qi, F. Li, and L. Zhenzhong, "Review on camera calibration," pp. 3354 – 3358, 06 2010.

[23] K. K. J. H. Vít Třebický, Jitka Fialová, "Focal length affects depicted shape and perception of facial images," *PLOS ONE*, 02 2016.

[24] R. Saborido Infantes, R. Morales, F. Khomh, Y.-G. Guéhéneuc, and G. Antoniol, "Getting the most from map data structures in android," *Empirical Software Engineering*, 03 2018.

[25] M. W. . M. K. . C. Marouane, "Indoor positioning using smartphone camera," *2011 International Conference on Indoor Positioning and Indoor Navigation*, november 2011.

[26] C. Thirumoorthi and K. Thirunavu, "Easy optimization of image transformation using sfft algorithm with halide language," 12 2014.

[27] N. Hui, L. Chieng, W. Ting, H. Mohamed, and M. Mohd Arshad, "Cross-platform mobile applications for android and ios," pp. 1–4, 04 2013.

[28] K. T. . W. D. Weiss, K., "A survey of transfer learning.," *J Big Data 3*, 05 2016.