DOI: 10.15649/2346030X.3789
Aibi revista de investigación, administración e ingeniería. Volume 12, Number 1, Pág 128-138, January - April 2024, ISSN 2346-030X

Deep learning-based obstacle detection in IoT framework for aiding visually impaired persons

Anamika Maurya[1], Prabhat Verma[2]
[1]Harcourt Butler Technical University Kanpur, Kanpur - India
ORCID: [1]*0009-0004-9604-221X*, [2]0000-0001-9193-7744

Received: August 26, 2023.
Accepted: November 12, 2023.
Published: January 01, 2024.

***Abstract*—** Visual impairment problems cannot be relieved completely, but with the assistance of advanced technology, their sufferings can be diminished by introducing obstacle detection approaches. In this paper, a new technique is devised, namely Henri Gas Solubility Optimization with Deep Convolutional Neural networks (HGSO+DeepCNN)-based obstacle detection for blind persons. The phases take place in obstacle detection involves Internet of Things (IoT) simulation, routing as well as obstacle detection. A simulated IoT network is composed of multiple nodes capable of gathering videos. The collected information is fed by means of optimization-based routing protocol, named HGSO, where fitness measures, namely distance, energy, and delay are selected. The obstacle detection from the video is carried out in IoT base station (BS) that involves several processing, namely extraction of video frame, feature extraction (FE), object detection (OD) and object recognition. A video frame conversion process converts the videos into multiple numbers of frames. Moreover, OD is executed depending on extracted powerful Convolutional Neural network (CNN) features through Deep joint segmentation. Finally, the obstacles that exist in the frame are detected using Deep CNN. The experimentation outcomes demonstrate that HGSO+Deep CNN model acquired accuracy, sensitivity, specificity, delay, energy and throughput of 0.959, 0.9683, 0.9987, 0.0236 sec, 0.3564 J and 0.7835 Mbps correspondingly.

Keywords: obstacle detection, henry gas solubility optimization, deep joint segmentation, deep convolutional neural network, cnn features.

*Corresponding author.
Email: cparul2605@gmail.com (Parul Choudhary).
Peer reviewing is a responsability of the Universidad de Santander.

How to cite this article: A. Maurya y P. Verma, "Deep learning-based obstacle detection in IoT framework for aiding visually impaired persons", *Aibi research, management and engineering journal*, vol. 12, no. 1, pp. 128-138 2024, doi: 10.15649/2346030X.3789

I. INTRODUCTION

IoT is one amongst important technologies that mainly engages with controlling, inter-linking, and controlling digital as well as electronic devices within the network through IP addresses. The development of higher-speed network inventions namely Long Term Evolution (LTE) as well as future 5G network technologies had been employed for simulating complete digital and electronic services as well as applications at fingertips. Though physically disabled persons especially blind people are not able to utilize innovative technologies and this makes them more distressed [1]. Visually impaired (VI) persons feel not easy to moving or traveling independently. Moreover, they are also very struggling to do regular activities and they require the help of friends or other members to move from one place to another [2], [3]. Visual impairment problems cannot be relieved completely, but with the assistance of advanced technology, their sufferings can be diminished. Most of the time visually disabled persons utilize a guide dog to do their routine moving activities [4], [5]. Hence, the object recognition methods are more useful to visually challenged people to recognize the objects. Thus, the OD approach provides a better solution for this issue [6], [7].

A Deep Learning (DL) architecture was employed for a visually disabled person, which utilized a late fusion of two CNN approaches. This fused approach was provided better results than some other existing approaches for activity identification [8]. The GoogLeNet and AlexNet of CNN approach complement each other in recognizing various features of similar class. To achieve this, the extracted frames from input video were fed into the GoogLeNet and AlexNet models, and the class score outcome was combined through Support Vector Machine (SVM). In [9], Recurrent Neural Network (RNN) follows the CNN for object identification [10]. Moreover, a softmax classifier was employed for object recognition, and the thresholding of the Hue, Saturation, and Intensity (HSI) model was employed for recognizing the color. The DL approaches were combined with computer vision methodologies for visually damaged outdoor navigation assistance in [10]. Another approach was utilized the regression-based mechanism to track the object without the need for additional information, utilizes quick camera movements for object recognition. An app for the smartphone users is intended to guide visually damaged persons in [11].

A foremost motivation is designing of HGSO+Deep CNN-based obstacle detection method for visually disabled people. Firstly, IoT node simulation is accomplished and videos are gathered for obstacle detection. Then, collected information is subjected using HGSO routing algorithm, where fitness value relies upon distance, delay and energy [12]. The obstacle recognition is executed in BS that conducts processes, namely video frame extraction, FE, OD, and obstacle detection. In BS, videos are transformed into frames and next, perform FE process for extracting CNN features [13]. After the features are extracted, then the objects are detected employing Deep joint segmentation process [14]. Finally, the obstacles are detected using Deep CNN thereby produces the output as chair, table, or any other.

A major involvement of this research is the modelling of the HGSO+Deep CNN method for VI people. HGSO algorithm is employed for broadcasting the data packet from one end to the other. The deep CNN method is used to detect the object from extracted frames.

The research paper is organized in the mentioned manner. Section 2 illustrates the literature survey of object detection method, section 3 depicts the system model, section 4 expounds the developed methodology, section 5 explains the experimentation of developed methodology, and section 6 describes the conclusion of the research.

Visually impaired persons feel not easy while shifting or traveling independently. Moreover, they are also very struggling to do regular activities, and require the help of friends or other members for moving from single place to other. Visual impairment troubles cannot be relieved completely, but with the assistance of advanced technology, their sufferings can be diminished. These motivate the researchers for researching this domain.

II. THEORETICAL FRAMEWORK

Hendra Kusuma., et al. [15] developed a deep learning scheme based on various facial expressions for VI people. This technique was effectively identified facial expression, while vocal communication. However, this method was failed to improve accuracy when gathering excess information in the dataset. Amey Hengle., et al. [16] developed the deep learning approach for VI people. This technique was maintained the visionless person up to date with the everyday news. This approach did not minimize computation cost. Priyan Malarvizhi Kumar., et al. [17] developed the Neural learning approach for recognizing the face from the images. This technique was provided precise messages in a format of audio to the visually challenged individuals such that communicate was performed effortlessly. This method failed to select biotic and abiotic objects in the environment. Ayesha Ashraf., et al. [18] developed the Sensitive smart stick empowered approach. Although the cost of computation of this approach was very less, it did not consider image processing techniques to recognize the objects.

The deep learning techniques were devised for effective obstacle recognition although this technique was failed to consider an aspect of facial languages, which can aid us in designing a facial appearance dataset for effective obstacle detection performance. Deep learning strategy was introduced for the obstacle detection performance, but this method was failed to select Document Image Analysis (DIA) for attaining more better outcome, and also this technique failed to offer multilingual support for the user to improve the performance. Sensitive smart stick empowered by IoT was established for visually impaired persons, but the major challenge lies in developing novel optimization algorithms to enhance the performance. A wearable vision assistance system was designed in, but the major challenge lies in using the other characters of binocular vision in the system for effective performance.

III. METHODOLOGY OR PROCEDURES

In this paper, HGSO+Deep CNN-based obstacle detection is introduced for visually impaired persons. The overall steps involved in the process of obstacle detection include IoT simulation, routing, and obstacle detection. Initially, the IoT network is simulated for collecting the videos. The collected videos are transmitted through the routing process, that is carried out with the proposed optimization algorithm, named HGSO [12], where the fitness measures, like distance, energy and delay are considered.

IoT network contains IoT nodes, gateway nodes and cluster head (CH). In an IoT network, diverse smart devices are associated with distributing data by means of optimum path. IoT devices are constructed using multiple models with a less powerful networking model. The lower power processing is conducted for the certain applications with extraordinary individuality

The processing steps involved in OD are video frame extraction, FE, object recognition and OD, which is performed in the BS. Here, input video is transformed into numerous frames using video frame extraction approach. Once after frames are extracted, then FE is done to takeout appropriate features from frames using CNN features. After that, the OD is completed with the aid of deep joint model. Finally, the object recognition method is performed to recognize the objects that exist in the frame using Deep CNN. Figure 1 illustrates block diagram of HGSO+Deep CNN-based obstacle detection method.

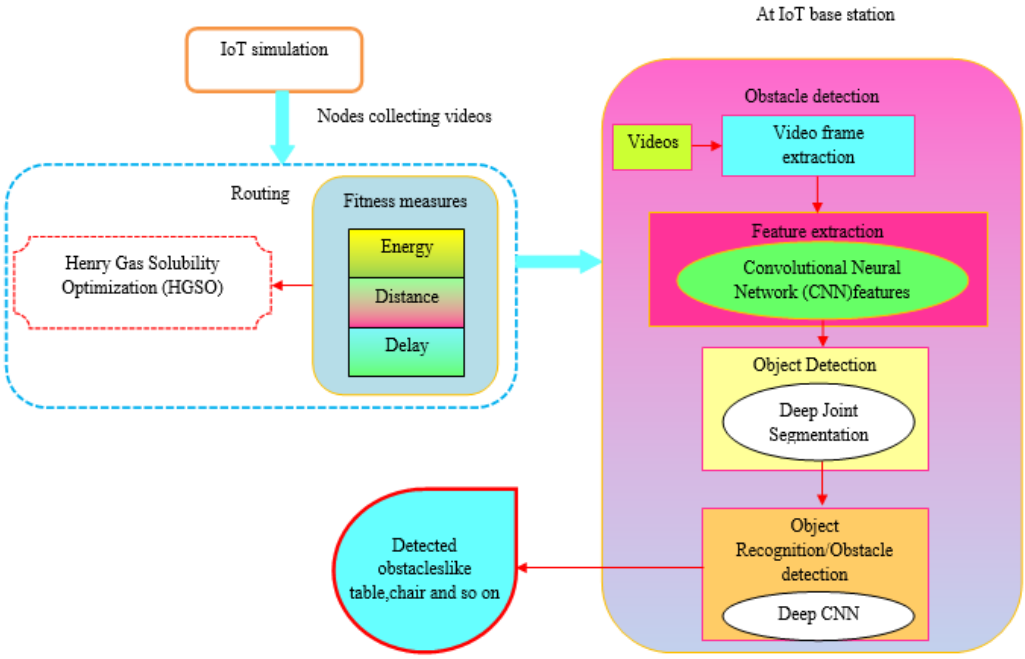

Figure 1: Schematic depiction of developed HGSO+Deep CNN-based obstacle detection.
Source: Own elaboration.

Firstly, the IoT nodes are created in a network for gathering videos. Consider S as total nodes in the IoT network, which is mathematically expressed as equation (1).

$$Z = \{Z_1, Z_2, \ldots, Z_e, \ldots, Z_s\} \quad (1)$$

Where, Z represents videos collected by the IoT nodes, s reveals overall nodes whereas Z_e represents video collected by e^{th} node that is selected to execute further procedure.

Routing using HGSO for obstacle detection

This section describes the HGSO-based routing algorithm for transmitting the transferred videos by means of optimum route. Firstly, nodes are devised, and videos gathered by the nodes are passed through optimum route utilizing HGSO routing approach. HGSO is designed by adapting the dissolving behaviour of gas in a given volume of a liquid at certain temperature. A benefit of using HGSO is that convergence speed as well as optimization performance of this approach is higher. While transmitting videos, an optimum route for transmission of data is determined using location update of the HGSO.

Solution Encoding

Solution encoding illustrates schematic representation of a solution vector for obstacle detection from videos. Furthermore, node having less distance, energy as well as delay is concerned as best outcome for obstacle detection. Figure 2 reveals solution encoding of the HGSO+Deep CNN algorithm. Here, s represents index of the node, where the optimal route is considered based on the node index s.

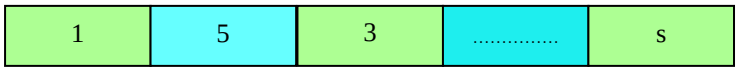

Figure 2: Solution encoding.
Source: Own elaboration.

Fitness Function

A fitness calculation is employed to select an optimum path using HGSO algorithm for forwarding the videos to the destination through the selected path. Depending on the minimum values of the fitness measure, like energy, distance, and delay, optimal route is chosen [14]. An expression for fitness function is expressed as equation (2).

$$R \sum_{\substack{p,q=1 \\ p \neq q}}^{s} \left(\frac{W + M_{p,q} + k}{3} \right)_{min} \quad (2)$$

The expression for delay and distance is described by equation (3) y (4)

$$M_{p,q} = \left\| p_p - p_q \right\| \quad (3)$$

$$k = \frac{o}{s} \quad (4)$$

where, Wrepresents the energy consumption, $M_{p,q}$reveals distance among $p^{th}$and q^{th} nodes, k depicts the delay, o indicates total nodes in a route, and $s$signifies overall nodes in network.

HGSO algorithm

HGSO is designed in terms of dissolving characteristics of gases in a given volume of the liquid. In this algorithm, the optimal path is determined by adapting the dissolving characteristics of gas. Initially, the quantity of gas and its position are initialized and then the gases are partitioned into the smaller cluster. Every cluster contains similar gases. From every cluster, the best gas is calculated, and then gases are arranged based on finest gas. Then, the henry coefficient, solubility, and location of gases are updated by estimating the optimal path. After updating the values, the gases are compared, and thereafter, finest gas is considered as optimum outcome. This process is continued till all the gases in the clusters are exhausted. In this manner, the best route is chosen. The locations update expression of the HGSO algorithm is given by equation (5).

$$A_{a,b}(s+1) = A_{a,b}(s) + Kw\sigma(A_{a,best}(s) - A_{a,b}(s)) + Kw\gamma(L_{a,b}(s)A_{best}(s) - A_{a,b}(s) \quad (5)$$

where, $A_{a,b}$ implies position of gas signified as $a$in the cluster$b$, implies randomly selected number, s implies iteration time, $A_{a,best}$specifies best gas $a$in cluster b, A_{best} indicates optimum gas in the swarm whereas σsignifies interaction capacity of the gas $b$in the cluster a, and$K$illustrates constant.

Herein, optimum path is determined by updating position of gas periodically depending on a fitness function, which involves energy, delay and distance. A formulation of fitness function is already explained in Eq. (2).

4.2 Obstacle detection using Deep learning

The obstacle detection from the video is accomplished in the BS that performs several processes namely video frame extraction, FE, OD, and object recognition or obstacle detection. The processing steps for detecting obstacles from the video are elaborated as follows.

Video frame extraction

A video frame extraction is employed to transforms the videos into multiple counts of frames since the further processing of this research is done based on the frames. The selected input video Z_ei s converted into n a number of frames, which is expressed aS equation (6).

$$Z_e = \{f_1, f_{2,}, , , f_v, \ldots, f_n\} \quad (6)$$

where, f_v represents the v^{th} frame that exists in the video, which is fed into the FE stage for further process.

An extracted frame f_v is forwarded to FE stage for taking CNN features from the extracted frames. An advantage of using the CNN feature is its providing effective performance with better convergence speed as it detects the obstacles more precisely. The CNN features are derived from the CNN network, which contains three layers, such as convolution (conv), fully connected (FC) and pooling (pool) layers. The extracted features are taken from FC layers. The extracted CNN features are indicated by $\Re$. An extracted CNN feature based on the conv layer is represented in figure 3.

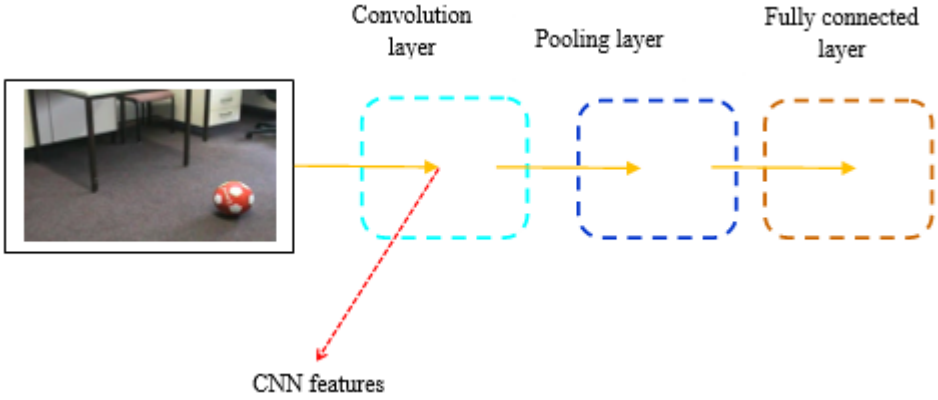

Figure 3. Extraction of CNN features
Sounrce: Own elaboration

OD using Deepjoint

An extracted feature implied by $\Re$ is passed to an input of Deep joint model for identifying the object from the extracted features. The Deep joint model [14] is comprised of three phases, like joining phase, region merging phase as well as generation of segmentation point model. Firstly, input images are fragmented into smaller grids and fused appropriate pixels using joining stage, which considers mean as well as threshold values. Thereafter completion of the joining phase, the appropriate areas is joined employing the bi-constraints. Moreover, region similarities are utilized for determining newer mean to estimate mapped points. Deep Joint model steps are given below.

Step 1: Configuration of Grids

At first, feature extracted image is partitioned into distinct smaller grids with grid size 2×2. The splitted grids are represented as equation (7).

$$\Re = \{Y_1, Y_2, \dots, Y_y, \dots, Y_l\} \qquad (7)$$

whereY_y signifiesy^{th} grid in the image, and $l$symbolizes overall grids.

Step 2: Joining phase

Once after grids are partitioned, pixels that exist in intra grid points are merged by comparing mean value with threshold value. A mean value is nothing but an average pixel value. The pixels in partitioned grid points are merged by comparing a mean value with threshold value, which is helpful to decide for fusing the grid point region. Hence, threshold is set as 1. An average value can be estimated by equation (8).

$$E = \frac{\sum_{d=1}^{Q} N_d}{Q} \qquad (8)$$

where,N_d represents pixel values connected to the grid E, and Q mentions total pixel in grid E. An equation for joining phase can be signified by equation (9).

$$E = \frac{\sum_{d=1}^{Q} N_d}{Q} \pm t \qquad (9)$$

where, t symbolize threshold value.

Step 3: Region merging phase

Herein, region merging matrix is made by adapting allotted grids. Once matrix is generated, region fusion is accomplished employing the resemblances of pixel intensity and likeness of bi-constraints for determining the similarity in an area. A region fusion is accomplished exploiting two criterions that should be satisfied, and is signified as:

(i) Mean value, stated asχ, must be smaller than 3.
(ii) From the entire grid, one grid point is selected.

From these two criterions, resemblance of an area is estimated, and these areas are joined for retrieving mapped points. A similarity region is signified as equation (10).

$$\chi = \frac{\sum_{m=1}^{U} N_x^y}{P} \qquad (10)$$

where, N_x^y depicts the combined pixels evaluated in combining phase in a specific grid Y_l, and Pdenotes overall combined pixels in Y_l. An integrated grid is specified as mapped points, which is revealed by equation (11).

$$J = J_1, J_2, \dots, J_g, \dots, J_h \qquad (11)$$

where, h symbolize a number of mapped points in image.

Step 4: Detection of deep points

The deep points are revealed by utilizing missed pixels. The residual pixels are known as missed pixels, and is stated as equation (12).

$$U = \{\mu_w\}; 1 < w \leq \kappa \qquad (12)$$

where, ν_w expresses the pixel, which is missed whereas κ implies overall missed pixels. therefore, deep points are calculated by totalling missed pixels as well as mapped pixels, which is denoted as equation (13).

$$\kappa_{po\,int\,s} = U + J_p \qquad (13)$$

where J_p represents the mapped pixels.

Step 5: Discovery of the optimal segment

Finally, finest segments are processed with the deep points by offering iterative approach. Firstly, segmented points signified by V are selected in random way. Hence, minimum distance is determined by equation (14).

$$S^{dist} = \sqrt{\sum_{p=1}^{r}(V_p - \kappa_{po\,int\,s_p})^2} \qquad (10)$$

The procedure is repeated and best-segmented points are chosen with MSE and assist to determine similarity amid two grids. Lastly, steps are continual till the ending state is achieved. An outcome attained from the deep joint is a table, chair, and so on, and the output vector of this classified output is represented as B_x.A pseudo-code of DeepJoint is explained in table 1.

Table 1: Pseudocode of DeepJoint model.

Input: Image $\mathfrak{R}$
Output: Detected Object
Dividing the image into $l$grids
Joining stage
For every grid in $\mathfrak{R}$
Combine the intra grid points using equation (12)
End for
Region merging stage
Estimate region similarity through equation (13)
Formulate the matrix of region fusion
if $\chi < 3$
Merge the regions with less χvalue to generate the mapped points
Else
Leave the region
end if
Deep point detection stage
Estimate the deep points through equation (16)
Randomly choose the detected points from deep points
for $d < d_{max}$
Estimate the distance among deep points using equation (17)
Predict the new detected points
Calculate MSE
End for
$n = n + 1$
Return the detected objects

Source: Obstacle detection using Deep CNN

This section depicts the obstacle detection using the DeepCNN classifier [23], which classifies the object like a table, chair, and so on.

Architecture of DeepCNN

A structure of the DeepCNN classifier is represented in figure 4. DeepCNN has three layers, namely conv, FC and pool layers, where individual layer executes various operations. Here, feature map is created based upon input vectorB_x. After formation of feature map, it is sampled, and thereafter subjected to pooling layer. Lastly, obstacle detection is carried out in the fully connected layer. Moreover, accuracy is enhanced relies on maximum count of Conv layers in the DeepCNN. The advantage of using DeepCNN is fast with effective classification performance. Figure 5 delineates structural illustration of DeepCNN.

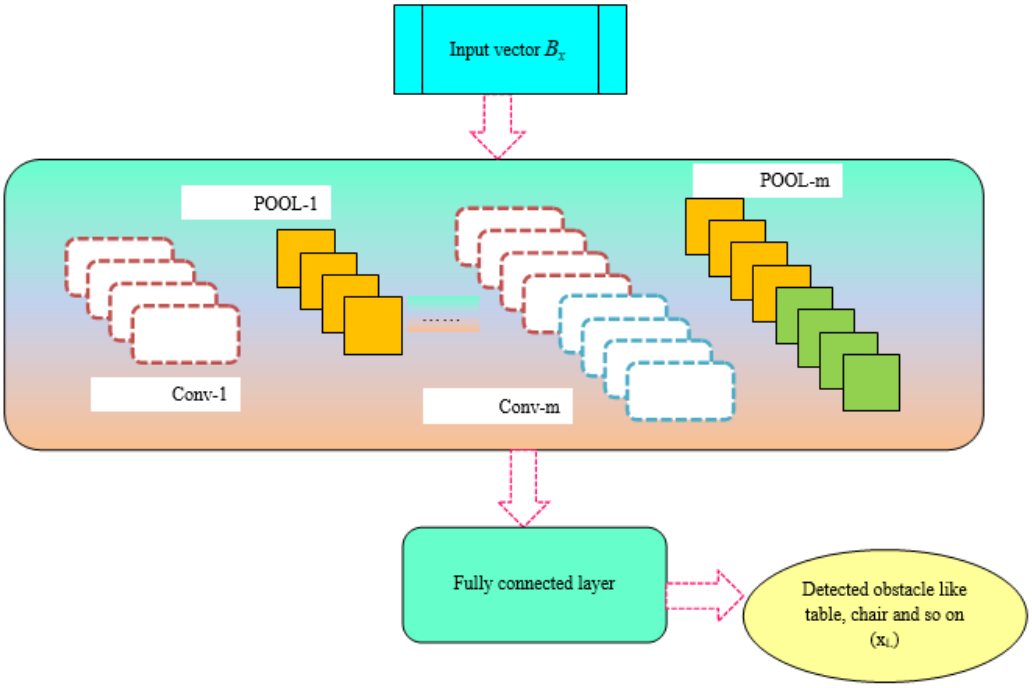

Figure 4: Structural illustration of DeepCNN.
Source: Own elaboration.

(i) Conv layers:

In the conv layer, a feature map is formed. Consider Deep CNN input by B_x, and an attained outcome from the Conv layer is specified as equation (15).

$$\left(B_x^{\,m}\right)_{c,a} = \left(b_x^{\,m}\right)_{c,a} + \sum_{i=1}^{l_1^{x-1}} \sum_{j=x_1^{\,x}}^{z_1^{\,x}} \sum_{\vartheta=z_2^x}^{z_2^{\,x}} \left(e_m^x, i\right)_{j,\vartheta} * \left(B_x^{\,m-1}\right)_{c+j,a+\vartheta} \quad (15)$$

where, $*$ depicts the conv operator, $(B_x^m)_{c,a}$ specifies stationary feature map or predicted resultant of x^{th} conv layers, that is cantered by(c, a) and $\left(B_x^{\,m-1}\right)_{c+j,a+\vartheta}$ e_m^x, i depicts kernel operation in x^{th} conv layers.

(ii) Pool layers: Pool layer is a non-parametric layer without weights or biases and thus, processes are successfully performed.

(iii) FC layers: Data collected from the pool layer is given as FC layer input to initiate classification of collected data. Lastly, output measured by means of FC layer can be formulated by equation (16).

$$k_x^m = \mu(F_x^{\,m}) with F_x^m = \sum_{i=1}^{l_1^{\,x-1}} \sum_{j=x_1^{\,x}}^{z_1^{\,x}} r \sum_{\vartheta=z_2^x}^{z_2^{\,x}} (e_m^x, i)_{j,\vartheta} * \left(B_x^{\,m-1}\right)_{c+j,a+\vartheta} \quad (16)$$

Hence, the output measured through DeepCNN is the table, chair, and other obstacles, which is represented asX^L.

IV. RESULTS AND DISCUSSION

This part reveals discussion of resultants based on HGSO+Deep CNN to detect obstacles from videos. The efficacy of HGSO+Deep CNN is estimated by comparing it with other conventional approaches.

Experimental setup and dataset description

The developed HGSO+Deep CNN approach is executed in Python tool with Kerasand Tensor flow using video object tracking dataset [19]. The video object tracking dataset contains 13 directories and 1 file. The dataset is prepared from 2D movies with fixed as well as moving cameras. Some of the directories that exist in the dataset are Babenko, Bobot, Kalal, Ross, Wang, Kwon, and so on.

Performance metrics

A performance improvement of HGSO+Deep CNN is assessed through metrics namely accuracy, specificity, energy, sensitivity, throughput and delay.

Accuracy: The term accuracy measures exactness of evaluated value with original value that is evaluated by equation (17).

$$X_a = \frac{\lambda_a + \lambda_b}{\lambda_a + \lambda_b + \mu_a + \mu_b} \quad (17)$$

where, λ_a signifies true positive, λ_b denotes true negative, μ_as ymbolizes false positive, and μ_bd epicts false negative.

Sensitivity: It is revealed as proportion of precisely recognized frames to overall frames that is stated by equation (18).

$$Y_{se} = \frac{\mu_b}{\mu_b + \mu_a} \quad (18)$$

Specificity: It is expressed as the proportion of correctly recognized frames to the total count of frames, which is stated as equation (19).

$$Z_{sp} = \frac{\mu_a}{\mu_a + \mu_b} \quad (19)$$

Delay: Delay mentions to the time assumed for forwarding the packets to destination and its unit is second.

Average Residual energy: It is stated as number of energy residues in a node after performing data transmission. The unit of energy is Joule.

Throughput: Throughput refers to the quantity of information effectively transported to the destination. The unit of throughput is represented as bits per second.

Simulation outcomes

Simulation outcomes of HGSO+Deep CNN algorithm are illustrated in this part. Figure 5 demonstrates experimental outcome of HGSO+Deep CNN algorithm for detecting obstacles from videos.

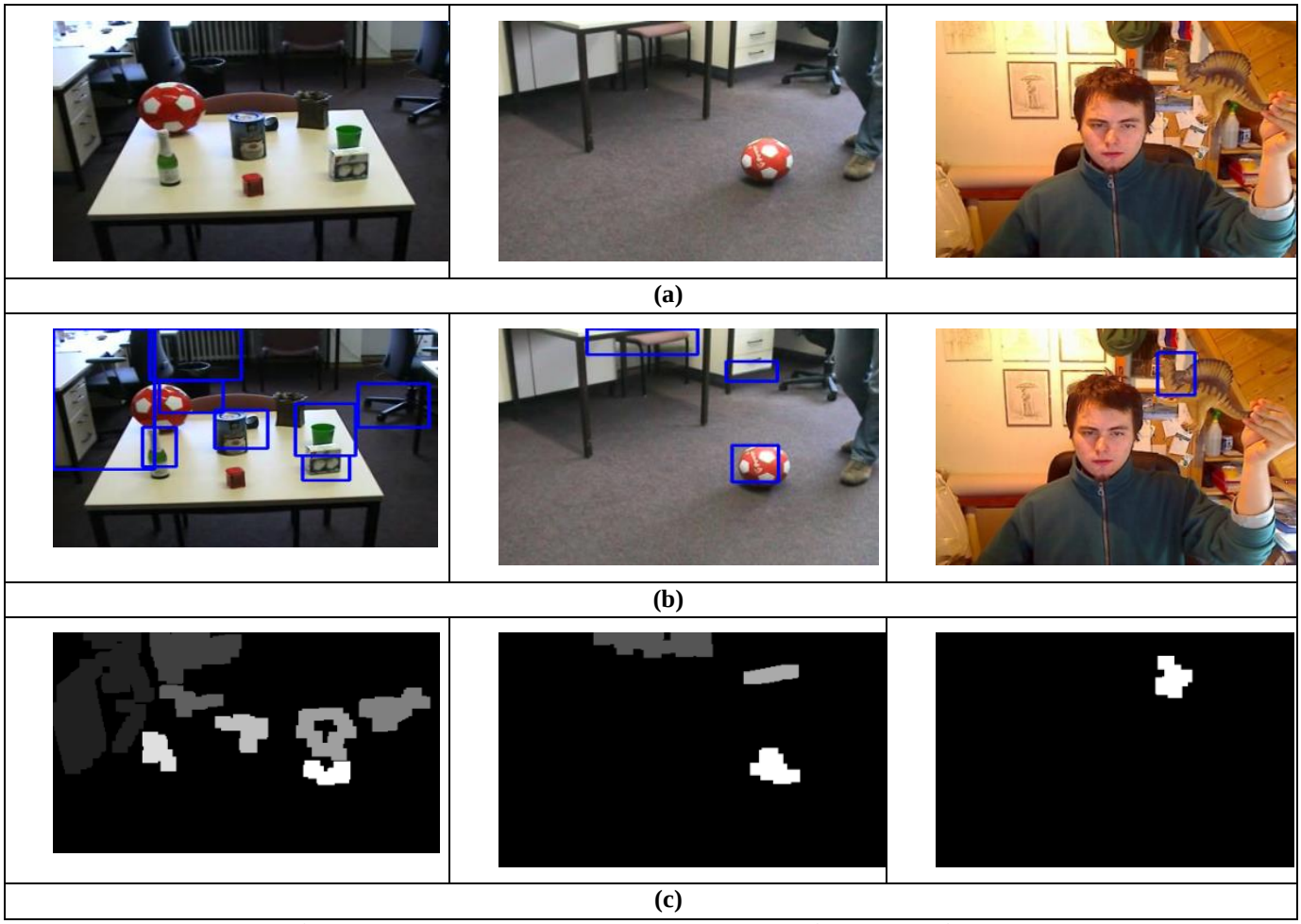

Figure 5: Experimental results of developed HGSO+Deep CNN algorithm based on a) Original frame extracted from the input video b) Detected object from the frame c) Recognized object from frame.
Source: Own elaboration.

Competing methods

The performance improvement of the developed HGSO+Deep CNN algorithm is analyzed by comparing the competing techniques, such as LEACH+LSTM [20], DSR+DisNet [21], and PSO+DNN [22].

Comparative assessment

A performance of HGSO+DeepCNN approach is assessed by altering training data and number of rounds.

Assessment regarding varying training data

Figure 6 reveals assessment of model relies on performance measures by changing training data. Figure 6 a) signifies estimation of accuracy. For considered training data is 80, accuracy achieved by HGSO+Deep CNN model is 0.957, whereas existing approaches, like LEACH+LSTM, DSR+DisNet and PSO+DNN accomplished the accuracy of 0.805, 0.864, and 0.901. Figure 6 b) delineates evaluation of sensitivity. For training data=90, sensitivity acquired through HGSO+Deep CNN is 0.968, whereas sensitivity obtained by LEACH+LSTM, DSR+DisNet, and PSO+DNN is 0.6436, 0.8988, and 0.8013. The improvement of performance attained by HGSO+DeepCNN based upon existing approaches is 28.48%, 7.17% and 17.24% while 90% training data. Figure 6 c) demonstrates an estimation of HGSO+DeepCNN concerning specificity. For training data is 70%, existing approaches, like LEACH+LSTM, DSR+DisNet and PSO+DNN attained specificity about 0.815, 0.833, and 0.925, and HGSO+Deep CNN acquired specificity of 0.979.

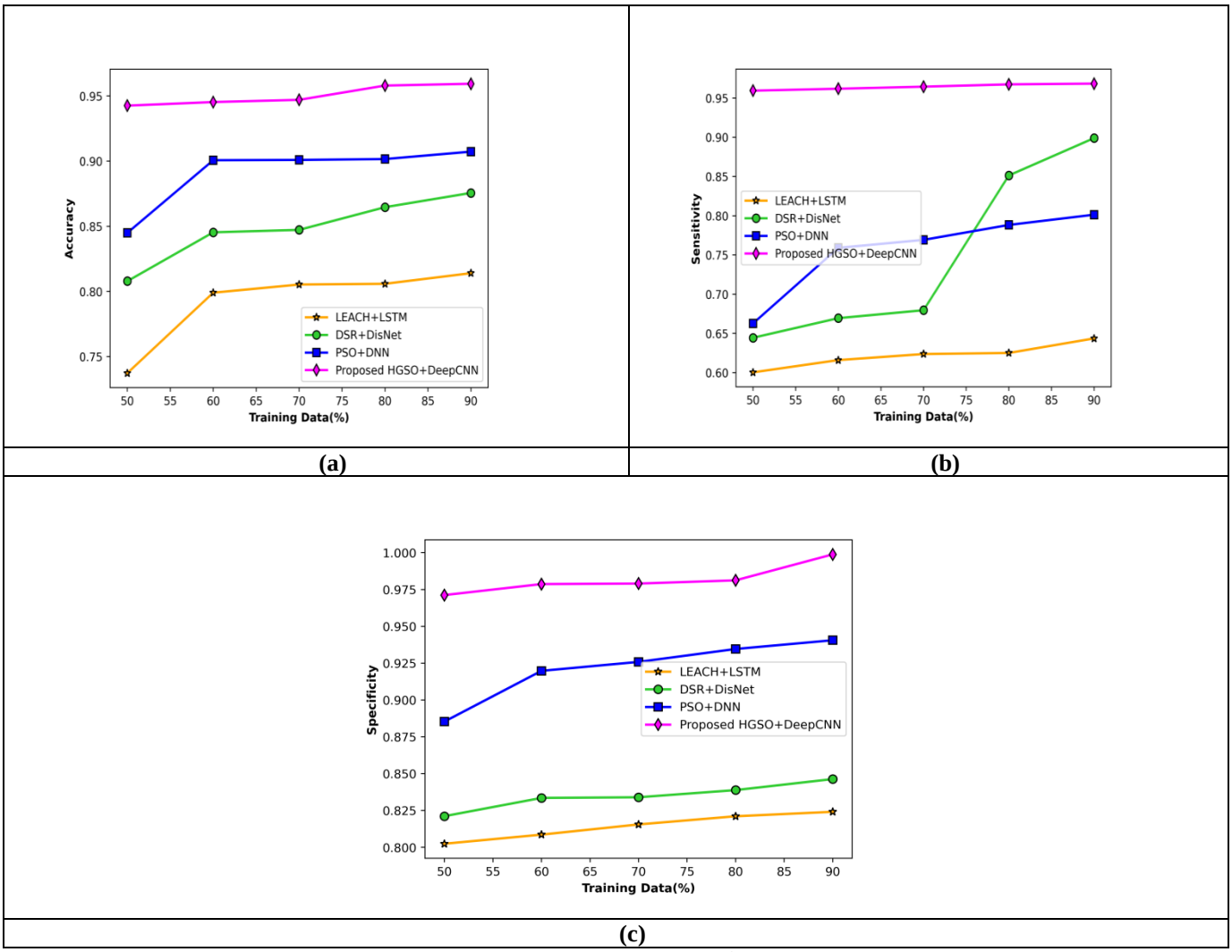

Figure 6. Evaluation by varying training data based on a) Accuracy, b) Sensitivity and c) Specificity.
Source: Own elaboration.

The percentage improvement achieved by the developed HGSO+Deep CNN model based on accuracy is 15.87%, 9.74%, and 5.88% for 80% training data.

The developed HGSO+Deep CNN method attained the percentage improvement of specificity is 16.69%, 14.81%, and 5.42% for 70% training data.

Thus we can see the HGSO + Deep CNN improves the performance of obstacles detection in terms of accuracy, sensitivity and specificity.

Evaluation based upon the varying count of rounds

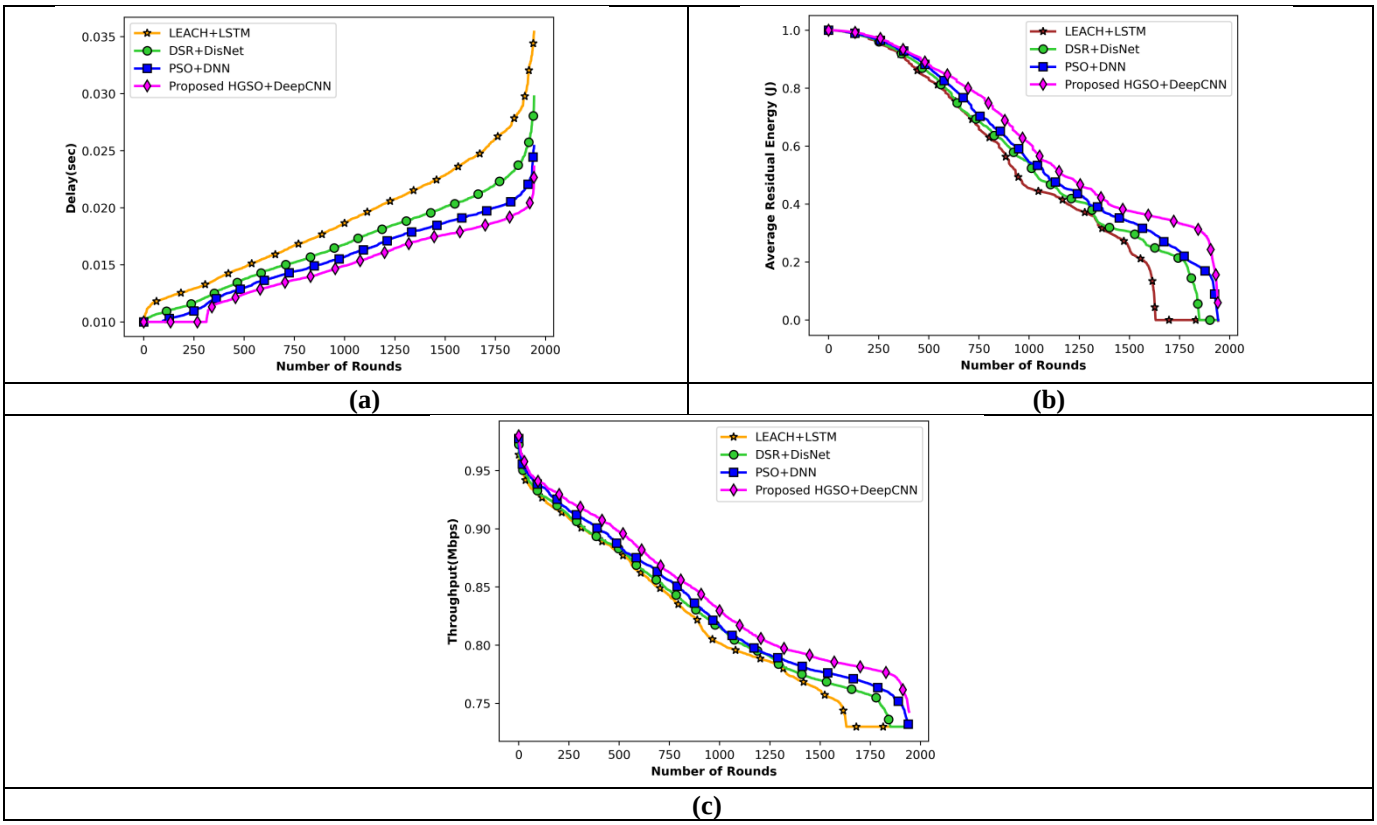

Figure7. Assessment by a varying number of rounds based on a) Delay, b) Average residual energy, and c) Throughput.
Source: Own elaboration.

Estimation of HGSO+Deep CNN is analyzed by altering number of rounds is given in figure 7. Figure 7 a) demonstrates evaluation based on delay. HGSO+Deep CNNmethod acquired the delay of 0.016 sec, while the existing approaches, such as LEACH+LSTM, DSR+DisNet, and PSO+DNN acquired delay of 0.020 sec, 0.018 sec, and 0.017 sec for 1200through. Figure 7 b) shows evaluation of energy by changing rounds. For round is 400, the developed Fractional SSA model acquired energy of0.924 J, and the existing approaches, like LEACH+LSTM, DSR+DisNet, and PSO+DNN acquired energy of 0.893 J, 0.906 J, and 0.917 J, correspondingly. Figure 7 c) displays estimation of throughput by changing rounds. When number of rounds=700, throughput measured by LEACH+LSTM, DSR+DisNet, and PSO+DNN is0.849 Mbps, 0.854 Mbps, and 0.861 Mbps. Similarly, Fractional-SSA measured throughput of 0.869 Mbps for round is 700.

Here the developed model shows the less delay while changing the no. of rounds as compare to existing models and higher throughput than LEACH+LSTM, DSR+DisNet and PSO+DNN.

Comparative discussion

Table 2. portrays the comparative values of HGSO+DeepCNN algorithm for detection of obstacle from videos. From the table, maximum accuracy acquired by HGSO+Deep CNN is 0.959, while LEACH+LSTM, DSR+DisNet, and PSO+DNN methods attained accuracy of 0.8140, 0.875, and 0.907 by changing training data percentage. HGSO+DeepCNN method attained maximal sensitivity of 0.9683, whereas the existing approaches, like LEACH+LSTM, DSR+DisNet and PSO+DNN acquired sensitivity about 0.6436, 0.8599, and 0.8013 by changing training data. Specificity measured by HGSO+DeepCNN is 0.9987, whereas LEACH+LSTM, DSR+DisNet, and PSO+DNN are 0.8241, 0.8463, and 0.9405 by varying training data percentage. Delay measured by HGSO+Deep CNN is 0.0236 sec, while LEACH+LSTM, DSR+DisNet, and PSO+DNN attained delay about 0.0354 sec, 0.0297 sec, and 0.0254 sec with a multiple numbers of rounds. The developed HGSO+DeepCNN residues energy of 0.3564 J, whereas LEACH+LSTM, DSR+DisNet, and PSO+DNN obtained energy of 0.032 J, 0.284 J, and 0.295 J by altering number of rounds. The throughput measured through the developed HGSO+DeepCNN method is 0.7835 Mbps, while the existing approaches, like LEACH+LSTM, DSR+DisNet, and PSO+DNN achieved the throughput of 0.7314 Mbps, 0.7634 Mbps, and 0.7725 Mbps by varying the rounds.

Table 2. Comparative Discussion.

Variations	Evaluation Metrics	LEACH+LSTM	DSR+DisNet	PSO+DNN	Developed HGSO+Deep CNN
By varying training data	Accuracy	0.814	0.875	0.907	**0.959**
	Sensitivity	0.6436	0.8599	0.8013	**0.9683**
	Specificity	0.8241	0.8463	0.9405	**0.9987**
By varying number of rounds	Delay (sec)	0.0354	0.0297	0.0254	**0.0236**
	Energy (J)	0.0320	0.2484	0.295	**0.3564**
	Throughput (Mbps)	0.7314	0.7634	0.7725	**0.7835**

Source: own elaboration.

V. CONCLUSIONES

This research introduced HGSO+DeepCNN to detect obstacles from the captured video. Firstly, simulation of IoT network is done for gathering videos. These videos are transferred through the optimal route, which is conducted by HGSO, where fitness measures, namely distance, energy, and delay are concerned. The phases executed in OD are video frame extraction, FE, OD, and object recognition, which is performed at BS. Herein, an input video is transformed into diverse frames using video frame extraction process. After frame extraction, then FE is accomplished for extracting suitable features from frames using CNN features. After that, the OD is done with the aid of Deepjointsmethod. Finally, the object recognition method is performed to recognize the objects that exist in the frame using Deep CNN. The experimentation outcomes demonstrate that HGSO+DeepCNN model acquired better performance regarding accuracy, sensitivity, specificity, delay, energy, and throughput of 0.959, 0.9683, 0.9987, 0.0236 sec, 0.3564 J, and 0.7835 Mbps, correspondingly. Future work will be performed by including some other effective optimization schemes.

VI. REFERENCES

[1] V. A. Vignesh and K. Madheswari, "Object detection application for visually challenged people using internet of things," International Journal for Research in Engineering Application & Management, vol. 2, no. 6, pp. 72–76, Mar. 2017.

[2] K. Potdar, C. D. Pai, and S. Akolkar, "A convolutional neural network based live object recognition system as blind aid,"Computer Vision and Pattern Recognition, 2018.

[3] S. Bhole and A. Dhok, "Deep Learning based Object Detection and Recognition Framework for the Visually-Impaired," in Proceedings of Fourth International Conference on Computing Methodologies and Communication (ICCMC), pp. 725–728, Mar. 2020.

[4] M. M. Islam, M. S. Sadi, K. Z. Zamli, and M. M. Ahmed, "Developing walking assistants for visually impaired people: A review," IEEE Sensors Journal, vol. 19, no. 8, pp. 2814–2828, Jan. 2019.

[5] R. Haque, M. Islam, K. S. Alam, H. Iqbal, and E. Shaik, A Computer Vision based Lane Detection Approach," International Journal of Image, Graphics & Signal Processing, vol. 11, no. 3, Mar. 2019.

[6] M. M. Islam, M. R. Islam, and M. S. Islam, "An efficient human computer interaction through hand gesture using deep convolutional neural network," SN Computer Science, vol. 1, no. 4, pp. 1–9, Jul. 2020.

[7] M. R. Puttaswamy, "Improved Deer Hunting Optimization Algorithm for video based salient object detection," Multimedia Research, vol. 3, no. 3, 2020.

[8] J. Monteiro, J. P. Aires, R. Granada, R. C. Barros, and F. Meneguzzi, "Virtual guide dog: An application to support visually-impaired people through deep convolutional neural networks," in 2017 International Joint Conference on Neural Networks (IJCNN), pp. 2267–2274, 2017.

[9] A. Renjit, "DeepJoint segmentation for the classification of severity-levels of glioma tumour using multimodal MRI images," IET Image Processing, vol. 14, no. 11, pp. 2541–2552, 2020.

[10] J. R. Tapu, B. Mocanu, and T. Zaharia, "Seeing without sight-an automatic cognition system dedicated to blind and visually impaired people," in Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 1452–1459, 2017.

[11] B.-S. Lin, C.-C. Lee, and P.-Y. Chiang, "Simple smartphone-based guiding system for visually impaired people," Sensors, vol. 17, no. 6, pp. 1371, 2017.

[12] F. A. Hashim, E. H. Houssein, M. S. Mabrouk, W. Al-Atabany, and S. Mirjalili, "Henry gas solubility optimization: A novel physics-based algorithm," Future Generation Computer Systems, vol. 101, pp. 646–667, Dec. 2019.

[13] F. Tu et al., "Deep convolutional neural network architecture with reconfigurable computation patterns," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 25, no. 8, pp. 2220–2233, 2017.

[14] R. Kumar and S. Meher, "A novel method for visually impaired using object recognition," in 2015 International Conference on Communications and Signal Processing (ICCSP), IEEE, pp. 772–776, 2015.

[15] H. Kusuma, M. Attamimi, and H. Fahrudin, "Deep learning based facial expressions recognition system for assisting visually impaired persons," Bulletin of Electrical Engineering and Informatics, vol. 9, no. 3, pp. 1208–1219, Jun. 2020.

[16] A. Hengle et al., "Smart Cap: A Deep Learning and IoT Based Assistant for the Visually Impaired," in 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), pp. 1109–1116, Aug. 2020.

[17] P. M. Kumar et al., "Intelligent face recognition and navigation system using neural learning for smart security in Internet of Things," Cluster Computing, vol. 22, no. 4, pp. 7733–7744, Jul. 2019.

[18] A. Ashraf, S. Noor, M. A. Farooq, A. Ali, and A. Hasham, "Iot Empowered Smart Stick Assistance for Visually Impaired People," International Journal of Science and Technology, vol. 9, no. 10, Oct. 2020.

[19] "Video object tracking dataset," [Online]. Available: https://www.kaggle.com/kmader/videoobjecttracking. [Accessed: May. 2021].

[20] R. Kumar and D. Kumar, "Multi-objective fractional artificial bee colony algorithm to energy aware routing protocol in wireless sensor network," Wireless Networks, vol. 22, no. 5, pp. 1461–1474, 2016.

[21] F. Z. Aadi and A. Sadiq, "Proposed real-time obstacle detection system for visually impaired assistance based on deep learning," International Journal of Advanced Trends in Computer Science and Engineering, vol. 9, no. 4, Jul. 2020.

[22] F. S. Bashiri, E. LaRose, J. C. Badger, R. M. D'Souza, Z. Yu, and P. Peissig, "Object detection to assist visually impaired people: A deep neural network adventure," in International Symposium on Visual Computing, pp. 500–510, Nov. 2018.

[23] J. C. Ying, C. Y. Li, G. W. Wu, J. X. Li, W. J. Chen, and D. L. Yang, "A deep learning approach to sensory navigation device for blind guidance," in Proceedings of IEEE 20th International Conference on High Performance Computing and Communications, IEEE 16th International Conference on Smart City, IEEE 4th International Conference on Data Science and Systems, pp. 1195–1200, Jun. 2018.