Computer Systems
Science and Engineering

Tech Science Press

DOI: 10.32604/csse.2023.036980
Article

Intelligent Deep Convolutional Neural Network Based Object Detection Model for Visually Challenged People

S. Kiruthika Devi[1], Amani Abdulrahman Albraikan[2], Fahd N. Al-Wesabi[3], Mohamed K. Nour[4], Ahmed Ashour[5] and Anwer Mustafa Hilal[6,*]

[1]Department of Computing Technology, SRM Institute of Science and Technology, India
[2]Department of Computer Sciences, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O.Box 84428, Riyadh, 11671, Saudi Arabia
[3]Department of Computer Science, College of Science & Art at Mahayil, King Khalid University, Saudi Arabia
[4]Department of Computer Science, College of Computing and Information System, Umm Al-Qura University, Saudi Arabia
[5]Department of Engineering Mathematics and Physics, Faculty of Engineering and Technology, Future University in Egypt, New Cairo, 11845, Egypt
[6]Department of Computer and Self Development, Preparatory Year Deanship, Prince Sattam bin Abdulaziz University, AlKharj, Saudi Arabia
*Corresponding Author: Anwer Mustafa Hilal. Email: a.hilal@psau.edu.sa
Received: 18 October 2022; Accepted: 21 December 2022

Abstract: Artificial Intelligence (AI) and Computer Vision (CV) advancements have led to many useful methodologies in recent years, particularly to help visually-challenged people. Object detection includes a variety of challenges, for example, handling multiple class images, images that get augmented when captured by a camera and so on. The test images include all these variants as well. These detection models alert them about their surroundings when they want to walk independently. This study compares four CNN-based pre-trained models: Residual Network (ResNet-50), Inception v3, Dense Convolutional Network (DenseNet-121), and SqueezeNet, predominantly used in image recognition applications. Based on the analysis performed on these test images, the study infers that Inception V3 outperformed other pre-trained models in terms of accuracy and speed. To further improve the performance of the Inception v3 model, the thermal exchange optimization (TEO) algorithm is applied to tune the hyperparameters (number of epochs, batch size, and learning rate) showing the novelty of the work. Better accuracy was achieved owing to the inclusion of an auxiliary classifier as a regularizer, hyperparameter optimizer, and factorization approach. Additionally, Inception V3 can handle images of different sizes. This makes Inception V3 the optimum model for assisting visually challenged people in real-world communication when integrated with Internet of Things (IoT)-based devices.

Keywords: Pre-trained models; object detection; visually challenged people; deep learning; Inception V3; DenseNet-121

1 Introduction

Visually-challenged persons experience many challenges in managing their day-to-day activities, such as reading, writing, road crossing, finding an object, etc. They use sticks and pet dogs to observe the surroundings and mobilize easily. Braille mechanism helps them in their reading and writing processes, while their mobility needs require attention from the research community [1]. Though several navigation models are available in the existing literature, there is a need to develop effective object detection models for visually-challenged people. The model should be developed so that it can detect and classify the objects under various constraint factors such as occlusion, scaled, blurred and illuminated nature under various environmental nature. The current research work attempts to compare and contrast the performance of four effective state-of-the-art models and find the best one that can detect objects under unfavourable and unpredictable scenarios [2]. Object detection is one of the challenges that need to be addressed. In this perspective, the current study compares the efficiency of four pre-trained models, ResNet-50, Inception v3, DenseNet-121 and SqueezeNet, in object detection from the images captured when visually-challenged people want to move independently. These four models have a proven track record of handling face recognition and image recognition tasks efficiently. In line with this, the current study attempts to analyze their capacity to handle multi-class objects which might be tilted, occlusive and scale-oriented [3].

In object detection and classification applications, manual extraction of features has been completely eradicated after the invention of Convolution Neural Networks (CNN) and powerful Deep Learning (DL) techniques. These techniques extract the features independently and detect and classify the objects [4]. Since a model is trained with many objects, developing a DL-based object detection technique tends to provide an accurate result. Internet of Things (IoT) devices, for instance, microcontrollers, can run neural networks efficiently. Thus, a deep learning model is integrated with an IoT system to develop it as an assistive device for visually-challenged people [5]. Numerous efficient Deep Learning-based object detection frameworks are available such as Single Shot Detector, You Only Look Once, AlexNet, Region-based CNN, its variants etc. Yet, there exist several challenges in the detection of objects in real-time scenarios under constrained conditions. In addition, several challenges are faced by the current Deep Learning framework works, such as object localization, handling of occlusion images, multi-class images, multi-scale images, 3-D images, detection of the region of interest in a crowded scene, etc. [6,7].

Transfer learning has proved itself as an efficient and fast technique in computer vision. This paper presents a robust deep learning model for object detection which can be integrated with any navigation assistance devices to be used by visually-challenged people in a real-time environment. The performance of ResNet-50, Inception v3, DenseNet-121 and SqueezeNet models, which are pre-trained with ImageNet-1000 dataset, was tested upon PASCAL dataset and compared. This comparison may provide an overview of how these pre-trained models work with object detection. Among the pre-trained models, Inception v3 is found to be better, and the performance can be further improved by using the thermal exchange optimization (TEO) algorithm as a hyperparameter optimizer. In short, the key contributions of the paper is given as follows.

- Develop a robust DL based object detection technique for visually-challenged people in a real-time environment.
- Examine the performance of different pre-trained models such as ResNet-50, Inception v3, DenseNet-121 and SqueezeNet models for feature extraction.
- Present TEO algorithm for the hyperparameter optimization of the DL models, showing the novelty of the work.

The rest of the paper is structured as follows; Section 2 details the literature survey that has been done on various object detection techniques. Section 3 describes the technical details of the deep learning architectures of all four pre-trained models. Then, Section 4 describes the experimental analysis and results. Finally, the conclusion of the experimental procedures and pointers for future work are discussed in Section 5.

2 Literature Review

The current section reviews the existing works conducted on object detection and classification using various machine learning algorithms and CNN-based deep learning algorithms, namely, ResNet-50, Inception v3, DenseNet-121 and SqueezeNet. Recent technical results have been achieved based on the latest developments in computer vision and Machine Learning (ML) methodologies. The methodologies designed for object detection to help visually-challenged people focus on direction-based access to freedom concerning mobility, orientation, barrier mitigation, and analysis. In a general context, the scenario deals with generating information regarding the environment and surrounding in textual format for visually-challenged people [8]. However, the methods still suffer from detecting complex scenarios that require multiple object detection.

Dilshad et al. [9] introduced an object detection framework that compares recorded images with the Query Image (QI). To compare the Query Image (QI) with previously saved images, the features are extracted using Scale Invariant Feature Transform (SIFT), Bag of Words (BoW), and Principal Component Analysis (PCA). Besides, hand-crafted features might not be sufficient for effective image description since the result is based on the judgment of representative images fed at the training time. Furthermore, the comparison of QI with reference images is computationally expensive. As an extension of this work, Dilshad et al. [10] proposed another framework that exploits Compressive Sensing (CS) to evaluate image representation and semantic similarity with the help of Gaussian Process Regression (GPR) methods. Malek et al. [11] utilized three low-level feature extraction models using predefined features that are unified using a Deep Learning approach, i.e., Auto Encoder Neural Network (AE) for high-dimension feature illustration. Single layer neural network was applied to map the AE features with multi-labelled images. Rapid development observed in computer vision these days is applied in object prediction and analysis, specifically for Convolutional Neural Networks.

In general, DL is a family of ML that is based on data representation. Therefore, the learning mechanism might get either supervised or unsupervised. Reputed and efficient deep structures are Stacked AE (SAE) [12], which is a product of integrating AEs, Deep Belief Networks (DBN), and CNN. Deep CNNs are reliable for accomplishing better results from applications such as image classification, object analysis, image segmentation, etc. Next, the eligibility of learning generic images is verified in comparison with previous techniques of hand-crafted properties. The newly deployed CNNs are embedded with convolution, pooling, and Fully Connected (FC) layers. Hence, the feature maps are produced from convolution layers and are projected towards nonlinear gating functions like the sigmoid function and advanced rectified linear unit (ReLU) function. Consequently, the final variables of the activation function are subjected to normalization, which further proceeds with generalization. The structure of CNN is trained using the Backpropagation (BP) approach [13]. Alternatively, the classification model computes and labels the mapping from input data in which the CNNs are prone to overfitting. Overfitting is nothing but accurately understanding the data. However, for unknown cases, it becomes a complicated process.

In general, CNNs are considered nonlinear models with many free weights to be learned. Following this, it exhibits flexibility in learning the data fed during training. Additionally, noisy data

is predicted when the unseen testing samples get influenced. Hence, with the help of limited training data, overfitting is assumed to be a vital problem in deep ML. Here, it is depicted that there is a need to forward CNNs for pre-trained auxiliary predictions using maximum data instead of computing the CNN training [14]. Rhyou et al. [15] developed a face recognition system for managing access control of a real-time authentication system using the ResNet model. This system predicts whether a person belongs to that organization or not based on the facial image database of their employees stored in their server. The model achieved 97% accuracy. But there is a need to develop a system to increase the accuracy.

Having discussed numerous approaches using machine learning and deep learning techniques for object detection, Transfer Learning (TL) is an alternative solution provider to scenarios in which the training is already done using benchmark datasets. Here, the testing can be directly done, thus reducing the computational costs. The current study explores four pre-trained models: ResNet 50, Inception v3, DenseNet-121 and SqueezeNet. These models have been well utilized in the image recognition arena. Zahid et al. [16] utilized the Inception V3 model for anomaly detection in surveillance cameras using Spatial and temporal feature extraction. Li et al. [17] DenseNet and Region Proposal Network models are used for object detection on PASCAL VOC and MS-COCO datasets. Alhichri et al. [18] used the SqueezeNet model for object detection to assist visually-challenged people in indoor environments, and the study used in-house datasets, namely (KSU1) and UTrento. Indoor navigation prediction was designed for visually-challenged people using sensors and deep learning algorithms on the IPIN2016 dataset. Several mechanisms have been proposed earlier to predict and classify objects per the literature review. Yet, there is a need to identify an accurate, cost-effective object classification model that can be integrated with assistance tools for visually-impaired people to achieve better navigation in unknown environments. The current research article compares the performances of four such pre-trained models for object detection. The performances of ResNet-50, Inception v3, DenseNet-121 and SqueezeNet pre-trained models were compared by testing the images captured from indoor as well as outdoor environments by the camera used for assisting visually-challenged people. Based on the performance outcomes, the current study intends to suggest a better model among the four models considered for the study so that it can be integrated with any assisting tool later. A detailed description of the experiment is given in the upcoming section.

3 Materials and Methods

To better understand the investigation, the technical details of the four pre-trained models, namely, the Deep Layered ResNet-50 model, complex heavily-engineered Inception v3 model, short connection DenseNet-121 model and a light-weighted SqueezeNet model, are briefly described in this section.

3.1 ResNet-50 Model

Deep neural network training is a time-consuming process, yet the trained model may, at times, be an overfit model too. At the time of training, a model with increased depth results in saturation of accuracy and degradation. To overcome these disadvantages, the Microsoft team has introduced a residual framework called ResNet. In the ResNet model, the residual block resolves degradations and gradient diminishing issues by skipping the training after a few layers [2]. These ResNet systems have shown optimal results for the ImageNet dataset [3] in the case of image classification tasks primarily. As shown in Fig. 1, a residual block represents how one or more than one layer get skipped at the time of training the network. The objective of skipping over layers is to elude the vanishing gradient problem by reusing activations from a previous layer until the adjacent layer learns the weights.

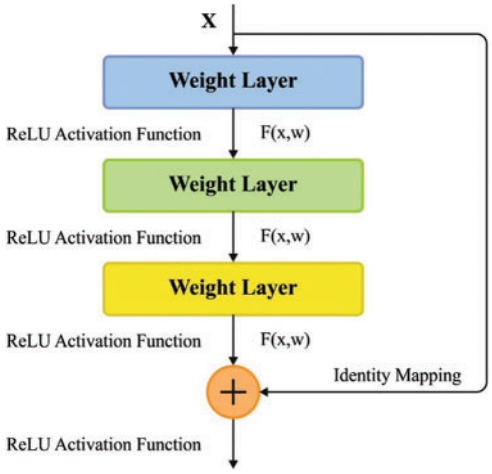

Figure 1: Single residual block

A residual expression is defined as $y = F(x, W) + x$, where x, W and y represent the residual block's input, weight, and output. ResNet system is composed of numerous residual blocks, and it is possible to add new layers to train the additional features if required. Many conventional ResNet structures exist, namely, ResNet-18, ResNet-50, and ResNet-101. In a current research paper, ResNet-50 is used, and the fundamental architecture of ResNet-50 is shown in Fig. 2. The extracted features are trained, and the training output is fed into the fully connected layer. This layer represents the last layer of ResNet, which is finally fed into the image classification layer.

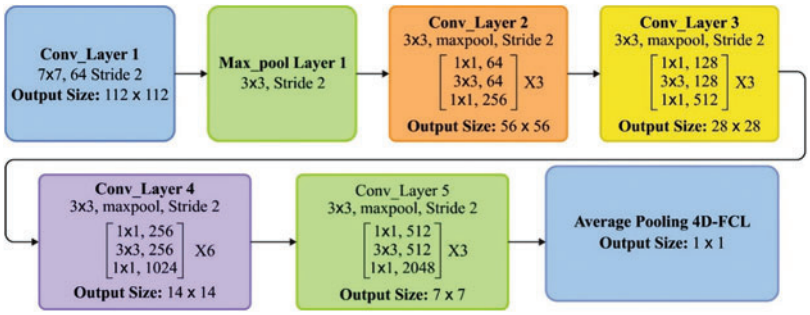

Figure 2: Layered structure of ResNet-50 model

3.2 Inception v3 Model

Inception V3 is an upgraded version of Google Net and is a classifier trained on the ILSVRC-2014 image dataset [4]. This model has a proven track record of accurate and speedy results. ResNet-50 tends to skip a few layers during training, while Inception v3 considers all its 48 layers yet achieves good accuracy and speed. This is attributed to an inception module in the latter, which is used to reduce the number of training parameters. The inception module reduces the parameters by factorization of the convolution layer. In addition, the presence of an auxiliary classifier layer integrated with central layers of the architecture helps the process as a regularizer [5]. Unlike classical CNNs like AlexNet and VGG, convolutional or pooling task is applied in the inception module and acquires benefits from every layer.

In addition, filters of different sizes have been applied in similar layers, which provide brief details and extract patterns from images of diverse sizes.

Fig. 3 shows the structure of Inception v3. The convolutional layer is referred to as the bottleneck layer. It is applied to reduce the parameters as well as computational complexity. Convolutional layers applied before a huge kernel convolutional filter helps in attaining good accuracy despite a reduced count of parameters. Furthermore, the convolutional layers tend to make the network deeper to add maximum non-linearity with the help of the ReLU activation function. Here, Fully Connected layers are replaced by the average pooling layer. As a result, the number of parameters decreases, while FC layers deal with many parameters. Therefore, it can learn in-depth representations of features with limited parameters compared to AlexNet.

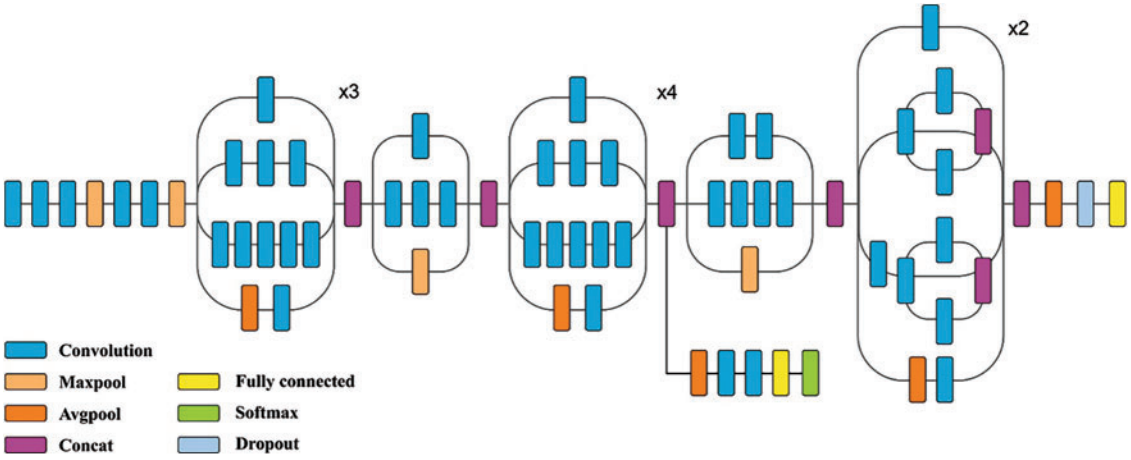

Figure 3: Structure of Inception v3 model

3.3 DenseNet-121 Model

This dense block is connected with many other dense blocks and forms the DenseNet model. Huang G proposed this structure [6] to reduce the parameters by limiting the connections between many layers. ResNet skips a few layer connections using a residual block, whereas a dense layer in a dense block obtains all the feature maps from every previous layer. Fig. 4 shows the layered structure of the DenseNet-121 model. A DenseNet consists of several dense blocks. A dense block contains multiple dense layers where the feature dimensions (such as width and height) remain the same within a dense block. Each dense layer of the dense block contains 1×1 convolution for feature extraction and 3×3 convolution for feature depth reduction with RELU & Batch Normalization (BN) as composite functions. The transition layer, in between the dense block with 1×1 convolution and 2×2 average pool layer, is used in the reduction of depth and size of the feature, respectively. The output of the last dense block is connected to a fully connected layer as a classifier and is used for prediction.

3.4 SqueezeNet Model

Fig. 5 depicts the structure of the SqueezeNet [7] model. Being a lightweight model, the SqueezeNet model is composed of convolution (conv), a Max-pooling layer with stride 2, fire modules (fire2-9), average pooling and a softmax layer as a classifier. It is capable of extracting the feature from low-resolution images. ResNet-50, InceptionV3 and DenseNet models detect the objects in two stages. First, the models locate the region of interest using a bound box and then classify the object. However,

SqueezeNet does both processes in a single stage, thus yielding high-speed detection. Also, this quality makes it highly suitable for high-dimensional image feature extraction.

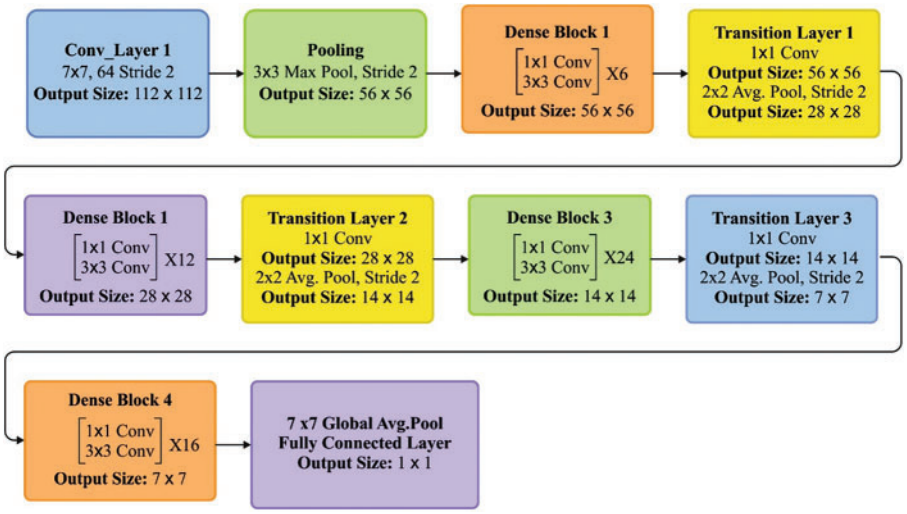

Figure 4: Layered structure of densenet model

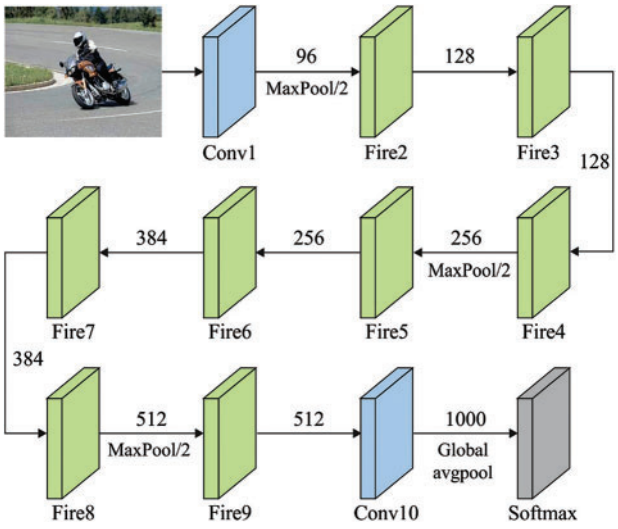

Figure 5: Structure of SqueezeNet

Fig. 6 shows the SqueezeNet model, whereas its fire module can shrink or expand the filter size based on the size of the images. Hence it is named after SqueezeNet. The squeezeNet model is also called as fire model. The fire module comprises two layers: the squeeze layer and the expand layer. The squeeze layer reduces the size of the feature map, whereas the expand layer improves the weight.

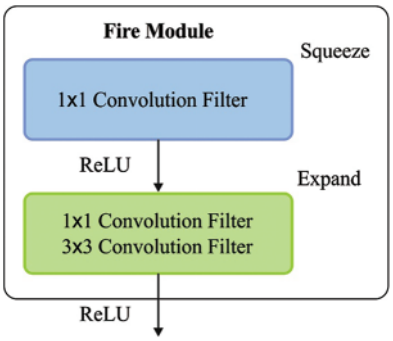

Figure 6: SqueezeNet fire module

3.5 Hyperparameter Tuning Model

The TEO algorithm can be employed to determine the DL models' hyperparameter values, such as batch size, learning rate, and the number of epochs, thereby improving the overall efficiency. The TEO approach depends on Newton's law of cooling. In the TEO approach, some agents are determined as the cooling object, and the rest represent the environment. Here, all the agents are considered as cooling objects, and by relating other agents as surrounding fluid, thermal exchange and heat transfer occur among them. This approach uses Newton's law of cooling to update the temperature. The algorithm is iterated till it satisfies the end condition [19].

The primary temperature of each object is defined in an m-dimension searching space.

$$T_i^0 = T_{\min} + rand \cdot . (T_{\max} - T_{\min}) \tag{1}$$

whereas T_i^0 represent the first solution vector of *the irh* object. Now $T_{\min}$, $T_{\max}$ denotes the bound of the design variable; random is a random vector within [0,1]; n represents the number of objects. Next, the objective function estimates the cost value of all the objects. Consider a memory that saves some historically best T vector, and the objective function value could enhance the efficiency of the approach without raising the computation cost. In this regard, a Thermal Memory (TM) is applied to save an optimal solution.

Next, the agent is classified into two different sets. For example, T_1 denotes an environment object for the $T_{\frac{n}{2}+1}$ cooling object. once an object has low β, then exchanges the temperature to some extent. The value of β for all the objects is estimated in the following,

$$\beta = \frac{Cost(object)}{Cost(worst\ object)} \tag{2}$$

Time is related to the number of iterations. The value of t for all the agents is estimated in the following,

$$t = \frac{iteration}{\text{Max } iteration} \tag{3}$$

The metaheuristic algorithm must be able to escape from the trap once the agent gets closer to a local optimal. The environment temperature changes, c_1 and c_2, denote the controlling variable.

$$T_i^{env} = (1 - (c_1 + c_2 \times (1 - t)) \times random.) \times T_i'^{env} \tag{4}$$

Here c_1 and c_2 denote the controlling variable. $T_i^{'env}$ represents the preceding temperature of an object that is altered to T_i^{env}. Based on the above steps, the new temperature of an object is upgraded as follows

$$T_i^{new} = T_i^{env} \cdot +(T_i^{old} - T_i^{env} \cdot) \exp(-\beta t) \times T_i^{'env} \tag{5}$$

The variable Pro within (0, 1) is presented and stated whether a component of the cooling object should be changed or not. For all the agents, Pro is compared to Ran(i) ($i = 1, 2, \ldots n$), which is a random value in the range of (0, 1). When Ran (i) < Pro, one dimension of the *ith* agent is randomly chosen in the following:

$$T_{i,j} = T_{j,\min} + rand \cdot (T_{j,\max} - T_{j,\min}) \times T_i^{'env} \tag{6}$$

whereas T_{ij} denotes the *jth* dimension of *the irh* agent. $T_{j,\min}$ and $T_{j,\max}$ correspondingly, denotes the lower and upper limits of the *jth* parameter. The optimization method would be terminated afterwards fixed amount of iterations. When the condition is not fulfilled, it returns to Step 2 for a new round of iteration, or else the algorithm would be stopped, and the optimal solution would be described.

4 Analysis of CNN-Based Pretrained Models

Mobilization is difficult for visually-challenged people in indoor and outdoor environments, even though several mobile application services and solutions have been in use to guide them. If the visually-challenged people's degree of mobility increases, irrespective of the environment, their quality of daily life quality also increases. New technological advancements such as IoT, image processing and computer vision can be brought together to develop a better assistance model. In the above aspect, object recognition and classification are vital tasks that need to be accomplished with the help of computer vision algorithms. The input for such algorithms may be collected through IoT-based devices such as sensors and cameras to capture a wide range of objects in real scenes. However, various objectives exist, and to recognize such huge varieties, the models have to learn the scenarios through deep learning algorithms for better prediction. The description of the mobilizing environment can be of voice feedback. Therefore, the images of an object can be captured in a real-time environment using different sorts of IoT devices. Further, the captured object images can be processed and classified using powerful deep-learning algorithms. The current research article compares the performance of such deep learning models.

CNN-based pre-trained deep learning models, ResNet-50, Inception v3, DenseNet-121 and SqueezeNet, were validated using PASCAL's dataset. PASCAL-VOC is a publicly-available dataset composed of annotated images containing object classes such as a person, vehicle, etc. This dataset is mainly used for image classification and object detection challenges. In the current study, PASCAL-VOC12, an updated version of the PASCAL-VOC dataset consisting of 11,530 images, was used to compare four pre-trained CNN-based models [20,21]. Image preprocessing directly impacts the accuracy of the model that has been trained and used. The current study used the pre-trained models, i.e., already trained on the ImageNet-1000 dataset. Those models were cross-evaluated on PASCAL VOC12 dataset images and the images downloaded from google search. At the time of evaluation, the images used for testing were resized 299 × 299 pixels automatically by pretrained models since this is the minimum need to fit the model.

CNN models were utilized to extract the feature vectors, which are then transformed to feature maps. The CNN model comprises multi-layered neurons that are meant to directly extract both low-level and high-level features from the pixels of the image with limited preprocessing. Feature extraction

occurs due to convolution layers and max pooling layers. The more challenging part of the CNN model is to design and arrange the convolution and max pool layers simply to extract significant, abstract and invariant features from the images for accurate classification. Subsequently, the extracted feature maps are fed into pre-trained models, and the objects are classified according to the object class confidence score. The performance of each model was tested using a wide range of objects, including single-image and multiple-object images, occluded images, scale-oriented images etc. Fig. 7 shows the workflow of comparison of CNN-based pre-trained models used for object detection, i.e., a traffic light image. Four different pre-trained models detect it, and the detection is portrayed using an object class confidence score, which is primarily a probability for each class. For instance, ResNet-50 shows the highest probability percentage of 89.57% for the traffic light class compared to the other classes predicted, namely, streetlight, pole, passenger and car.

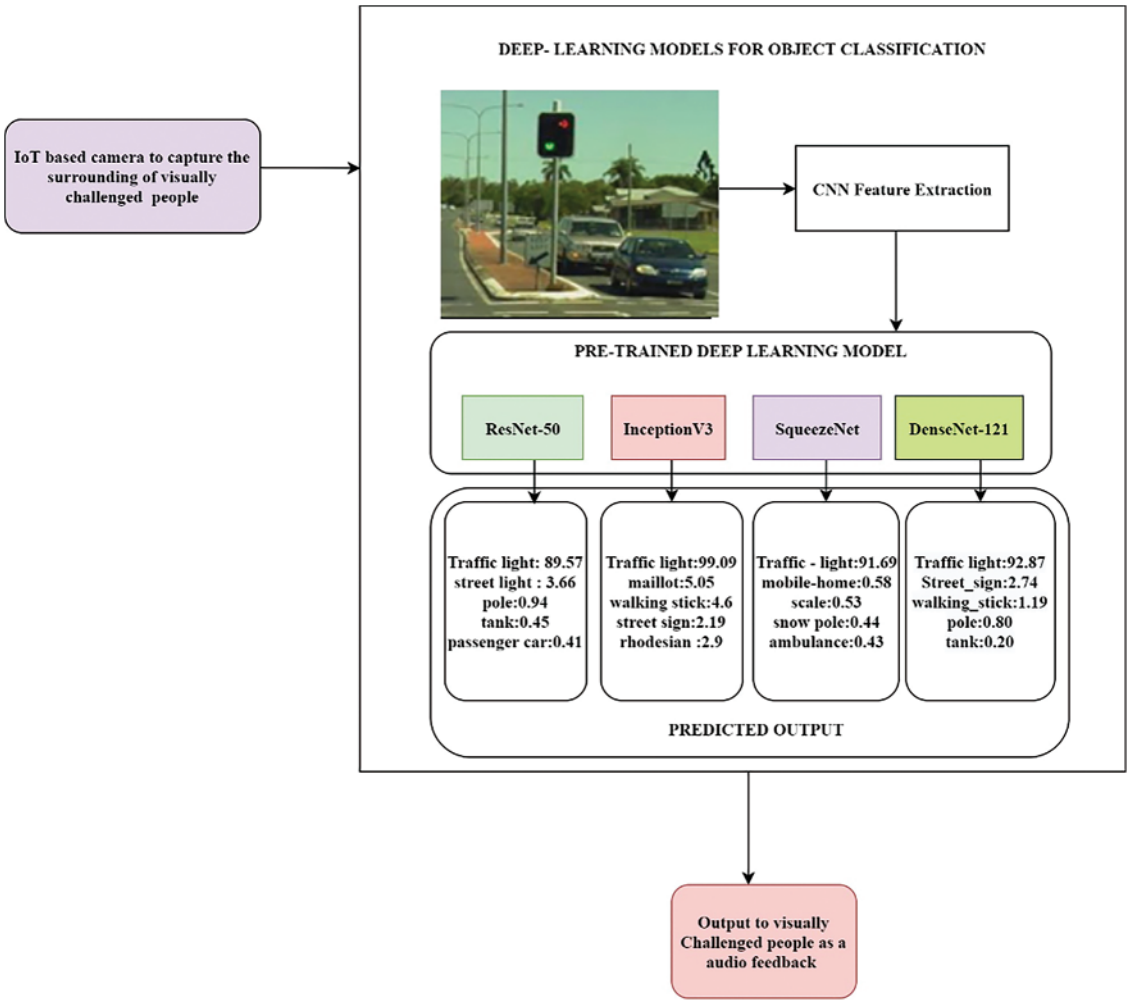

Figure 7: Workflow for the comparison of IoT integrated CNN-based pre-trained models for object detection

Table 1 portrays the results of detection performance achieved by four CNN-based pre-trained models for a few test images. In the case of the 'shopping cart' image, the SqueezeNet model wrongly detected it as a 'hamper' image, and its detection accuracy was 21.22%. While, ResNet-50, Inception v3, and DenseNet-121 models rightly classified the object as a 'shopping cart' with detection accuracy values being 86.91%, 96.97% and 66.20%, respectively. Similarly, the models also predicted various test images and the results are shown.

Table 1: Result analysis of different pre-trained deep learning models on object recognition

Image name	Pre-trained deep learning models							
	ResNet-50		Inception v3		SqueezeNet		DenseNet-121	
	Classified object	OCS %	Classified object	OCS %	Classified object	OCS %	Classified object	OCS %
Tea pot	teapot	97.79	teapot	99.96	teapot	96.96	teapot	99.89
	coffeepot	2.18	coffeepot	0.04	water_jug	3.04	coffeepot	0.08
	water_jug	0.02	water_jug	0.00	pitcher	0.00	water_jug	0.02
	pitcher	0.00	basenji	0.00	coffeepot	0.00	basenji	0.01
	espresso_maker	0.00	strainer	0.00	whiskey_jug	0.00	espresso_maker	0.00
Barber chair	barber_chair	70.68	moped	44.57	barber_chair	59.21	barbershop	52.45
	barbershop	28.80	barbershop	28.45	barbershop	34.02	barber_chair	33.75
	jinrikisha	0.14	barber_chair	23.75	shoe_shop	1.16	motor_scooter	13.21
	pay-phone	0.08	motor_scooter	3.21	motor_scooter	0.98	moped	0.40
	cash_machine	0.03	jinrikisha	0.02	unicycle	0.67	jinrikisha	0.10
Trolley bus	trolleybus	95.77	trolleybus	99.26	trolleybus	85.12	trolleybus	98.26
	passenger_car	3.58	passenger_car	0.71	passenger_car	6.36	passenger_car	1.71
	minibus	0.43	minibus	0.01	minibus	5.57	minibus	0.01
	streetcar	0.17	school_bus	0.00	streetcar	1.12	school_bus	0.00
	school_bus	0.02	recreational_ vehicle	0.00	police_van	1.08	police_van	0.00
School bus	school_bus	93.03	school_bus	99.99	school_bus	57.43	school_bus	99.00
	passenger_car	4.48	minibus	0.00	crane	33.94	passenger_car	1.00
	trolleybus	1.52	passenger_car	0.00	minibus	4.03	toucan	0.00
	streetcar	0.33	toucan	0.00	passenger_car	2.03	trolleybus	0.00
	moving_van	0.25	trolleybus	0.00	moving_van	1.48	minibus	0.00
Shopping cart (Augmented)	fire_screen	22.21	racket	74.15	scabbard	12.77	spider_web	12.21
	spider_web	5.70	volleyball	6.10	weevil	9.92	window_screen	10.70
	window_screen	5.13	safety_pin	2.33	chain	5.38	bow	2.33
	bow	4.75	tennis_ball	2.29	scorpion	4.82	scabbard	2.29
	racket	4.48	shopping_cart	2.14	screw	3.47	weevil	2.14
Shopping cart	shopping_cart	86.91	shopping cart	96.97	hamper	21.22	shopping_cart	66.20
	parachute	1.82	shopping_basket	3.02	handkerchief	16.39	parachute	21.56
	hamper	1.64	grocery_store	0.00	vase	8.59	shopping_basket	11.34
	shopping_basket	1.44	plate_rack	0.00	shopping_cart	5.60	grocery_store	0.71
	broom	1.32	bassinet	0.00	plastic_bag	4.96	plate_rack	0.13

(Continued)

Table 1: Continued

Image name	Pre-trained deep learning models							
	ResNet-50		Inception v3		SqueezeNet		DenseNet-121	
	Classified object	OCS %	Classified object	OCS %	Classified object	OCS %	Classified object	OCS %
Switch	switch	93.87	switch	97.36	switch	55.82	switch	85.91
	combination_lock	1.74	computer_keyboard	1.60	wall_clock	17.27	wall_clock	2.82
	safe	1.19	vending_machine	0.65	safe	5.51	combination_lock	1.33
	medicine_chest	0.80	oscilloscope	0.11	combination_lock	5.42	safe	1.45
	cassette	0.45	tape_player	0.07	analog_clock	3.36	medicine_chest	1.32
Window shade	window_shade	61.94	window_shade	83.96	window_screen	38.35	window_shade	66.94
	window_screen	9.97	medicine_chest	10.46	window_shade	15.39	window_screen	6.97
	mobile_home	8.14	mobile_home	4.57	bannister	13.11	mobile_home	4.14
	medicine_chest	5.41	window_screen	0.52	sliding_door	6.66	window_screen	3.41
	passenger_car	1.45	sliding_door	0.18	medicine_chest	4.99	sliding_door	1.45
Traffic signal	traffic_light	89.58	traffic_light	99.78	traffic_light	91.69	traffic_light	92.87
	street_sign	3.66	maillot	0.00	mobile_home	0.58	street_sign	2.74
	pole	0.95	walking_stick	0.00	scale	0.53	walking_stick	1.19
	tank	0.46	street_sign	0.00	snowplow	0.44	pole	0.80
	passenger_car	0.41	Rhodesian_ridgeback	0.00	ambulance	0.43	tank	0.20
Bannister	bannister	70.05	bannister	59.58	bannister	69.97	bannister	65.82
	prison	6.33	library	29.66	crate	5.82	library	7.27
	street_sign	2.16	bookcase	3.57	prison	4.27	street_sign	6.51
	palace	2.03	bookshop	3.54	coil	2.42	bookshop	3.42
	moving_van	1.58	barber_chair	0.65	photocopier	2.29	prison	3.36
Bannister (occluded image)	bannister	97.79	bannister	99.83	bannister	20.19	bannister	93.79
	coil	1.14	coil	0.18	upright	8.56	coil	3.14
	prison	0.21	rocking_chair	0.03	pedestal	6.27	pedestal	2.21
	window_shade	0.16	park_bench	0.00	photocopier	5.92	prison	0.16
	bookcase	0.11	folding_chair	0.00	bath_towel	4.38	rocking_chair	0.11

4.1 Result Analysis on PASCAL VOC12 Dataset

This section examines the performance of four models shown in terms of the mean Average Precision (mAP) metric on PASCAL VOC12. The benchmark dataset PACAL VOC12 contains a total of 20 object classes which are broadly categorized as a person, animal, vehicle and indoor category. The formula to calculate mAP, a measure of the model's performance, is given in Eq. (7). Table 2 provides an overall performance of the pre-trained deep learning models on the PASCAL VOC-12 dataset. Further, the formulae to calculate interpolated Average Precision (AP), Precision and Recall are depicted in Eqs. (7)–(9).

$$mAP = \frac{1}{n}\sum_{k=1}^{k=n} AP_k \tag{7}$$

where AP_k, denotes the average precision of class 'k' and 'n' corresponds to the number of classes.

$$AP = \frac{1}{11} \sum_{Recall \in \{0.0, 0.1, \ldots, 1.0\}} \left(Precision_{interpolation}(Recall) \right) \tag{8}$$

$$Precision = \frac{True\ positive}{True\ positive + False\ positive} \tag{9}$$

Table 2: Performance pre-trained deep learning models on the PASCAL VOC-12 dataset

Pre-trained methods	Mean average precision (mAP)			
	PASCAL VOC12 dataset classes			
	Person	Animal	Vehicle	Indoor
ResNet-50 model	82.90	75.08	76.23	75.60
Inception v3 model	80.92	94.34	95.60	88.50
DenseNet-121 model	78.30	67.69	87.65	79.50
SqueezeNet model	49.50	78.25	80.75	52.46

Table 3 provides a comparative map analysis of the Inception v3 model and the TEO-Inception v3 model. The results demonstrated that the Inception v3 model had obtained mAP of 80.92%, 94.34%, 95.60%, and 88.50% on detecting person, animal, vehicle, and indoor objects, respectively. However, the TEO-Inception v3 model has resulted in effective outcomes with increased mAP values of 85.87%, 97.09%, 96.31%, and 94.46% under person, animal, vehicle, and indoor, respectively.

Table 3: mAP analysis of Inception v3 and TEO-Inception v3 model on PASCAL VOC12 dataset

Pre-trained methods	Mean average precision (mAP)			
	PASCAL VOC12 dataset classes			
	Person	Animal	Vehicle	Indoor
Inception v3 model	80.92	94.34	95.60	88.50
TEO-Inception v3	85.87	97.09	96.31	94.46

4.2 Result Analysis on Google Search Images Dataset

Table 4 provides a brief comparative mAP examination of the TEO-Inception v3 model on the Google search image dataset. The results indicated that the SqueezeNet model had obtained lower mAP values over the other methods. At the same time, the ResNet-50 and DenseNet-121 models have resulted in moderately closer mAP. However, the Inception v3 model has accomplished maximum mAP values of 83.06%, 81.78%, 82.25%, and 83.48% under person, animal, vehicle, and indoor class objects, respectively.

Since the Inception v3 model has outperformed the other pretrained DL models, another comparative analysis with the TEO-Inception v3 model has been performed in Table 5. The experimental values indicated that the TEO-Inception v3 model had outperformed the Inception v3 model with the mAP of 87.28%, 88.45%, 91.93%, and 90.95% under person, animal, vehicle, and indoor class

objects respectively. The improved performance of the TEO-Inception v3 model is due to the unique characteristics of Inception v3 and the hyperparameter optimization process.

Table 4: Performance pre-trained deep learning models on google search images dataset

Pre-trained methods	Mean average precision (mAP)			
	Google search images			
	Person	Animal	Vehicle	Indoor
ResNet-50 model	73.98	75.06	75.22	74.99
Inception v3 model	83.06	81.78	82.25	83.48
DenseNet-121 model	75.18	77.30	75.20	78.29
SqueezeNet model	67.10	70.05	69.23	71.09

Table 5: mAP analysis of Inception v3 and TEO-Inception v3 model on google search images dataset

Pre-trained methods	Mean average precision (mAP)			
	Google search images			
	Person	Animal	Vehicle	Indoor
Inception v3 model	83.06	81.78	82.25	83.48
TEO-Inception v3	87.28	88.45	91.93	90.95

4.3 Discussion

Finally, a detailed overall mAP analysis of the pretrained DL models and TEO-Inception v3 model on the PASCAL VOC-12 and Google Search Images datasets are offered in Table 6. The results show that the SqueezeNet model has resulted in the least mAP values of 65.24% and 69.37% on the PASCAL VOC-12 and Google Search Images datasets, respectively. Followed by the ResNet-50 and DenseNet-121 models have resulted in moderately improved values of mAP. Also, the Inception v3 model exhibits better performance over the other pretrained models, with an overall mAP of 89.84% and 82.64% on the PASCAL VOC-12 and Google Search Images datasets, respectively. However, the TEO-Inception v3 model has outperformed the other models with the maximum mAP of 93.43% and 89.65% on the PASCAL VOC-12 and Google Search Images datasets, respectively.

It can be inferred from the above-mentioned tables and figures that the Inception V3 model is a superior performer in object detection and classification tasks than the other three models since it attained a high mAP of 89.84% on the PASCAL VOC12 dataset and 82.64% on the test images downloaded from Google image search. From the graphical representations of mAP attained by four different pre-trained models tested upon the PASCAL VOC12 dataset and test images downloaded from Google image search. It can be inferred that the SqueezeNet model achieved the least mAP % in object detection than the rest of the models compared. This is because SqueezeNet cannot handle oriented and occluded images, as the model is not trained with these features. Though the ResNet-50

model outperformed the SqueezeNet model, it was unable to achieve better results than DenseNet-121 and InceptionV3 models. ResNet-50 model is found to be good at facial recognition. However, it suffers during the classification of multi-class images.

Table 6: Overall mAP analysis on PASCAL VOC12 and google search images dataset

Pre-trained methods	Mean average precision (mAP)	
	PASCAL VOC-12	Google search images
ResNet-50 model	77.45	74.81
Inception v3 model	89.84	82.64
DenseNet-121 model	78.28	76.49
SqueezeNet model	65.24	69.37
TEO-Inception v3	93.43	89.65

The denseNet-121 model yielded a good performance, yet it failed to outperform the detection performance of the Inception V3 model. Dense-net 121 model achieved better performance, while its performance can further be enhanced through hyperparameter tuning and more training datasets. Therefore, the Inception V3 model is found to be an effective object detection model for the images captured for assisting visually-challenged people since it is flexible in recognizing images of various sizes and possesses high effectiveness due to a minimal number of parameters. Inception V3 is 42 layers deep and involves factorizing convolution layers, which reduces parameters. This characteristic makes the model predict and classify the object faster without any drop in efficiency. The improved accuracy of the Inception v3 model is due to the better learning nature and the inclusion of an auxiliary classifier. Inception V3 architecture also includes efficient reduction of grid size, which in turn leads to low computational cost. Hence the model proves to be highly efficient than the compared models in terms of speed, accuracy and cost. The inception V3 model can be integrated with object detection tools, supported by the IoT framework, to assist visually-challenged people [22].

5 Conclusion

The current research attempted to compare four different CNN-based object detection models: ResNet 50, Inception v3, DenseNet-121, and SqueezeNet. An extensive experimental analysis was conducted on the PASCAL VOC12 dataset and 100 images downloaded from Google image search. These images' characteristics match those of the images captured in the camera used for assisting visually-challenged people. The experimental results inferred that Inception V3 outperformed other CNN-based deep learning models in terms of accuracy. This is due to the peculiar nature of the Inception V3 architecture and TEO-based hyperparameter tuning process. This model can be easily integrated with tools that assist visually-challenged people. The rest of the CNN models, such as ResNet-50, DenseNet-121 and SqueezeNet, can also perform on par with Inception V3. This is possible because these models are trained with the necessary features. In future, efficient image preprocessing techniques can be included to embed with a navigation assisting tool. The current research attempted to find a better deep learning model among the selected four models by comparing their performances. In future, the better performer i.e., the Inception V3 model, can be integrated with an IoT-based assisting tool to help the visually-challenged people.

Funding Statement: Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2023R191), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. The authors would like to thank the Deanship of Scientific Research at Umm Al-Qura University for supporting this work by Grant Code: (22UQU4310373DSR61). This study is supported via funding from Prince Sattam bin Abdulaziz University project number (PSAU/2023/R/1444).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] S. S. A. Zaidi, M. S. Ansari, A. Aslam, N. Kanwal, M. Asghar *et al.,* "A survey of modern deep learning based object detection models," *Digital Signal Processing*, vol. 126, no. 11, pp. 103514, 2022.

[2] Z. Wu, C. Shen and A. van den Hengel, "Wider or deeper: Revisiting the resnet model for visual recognition," *Pattern Recognition*, vol. 90, no. 3, pp. 119–133, 2019.

[3] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *ACM Communications*, vol. 60, no. 6, pp. 84–90, 2017.

[4] F. Shao, L. Chen, J. Shao, W. Ji, S. Xiao *et al.,* "Deep learning for weakly-supervised object detection and localization: A survey," *Neurocomputing*, vol. 496, pp. 192–207, 2022.

[5] W. H. Bangyal, K. Nisar, A. A. B. A. Ibrahim, M. R. Haque, J. J. P. C. Rodrigues *et al.,* "Comparative analysis of low discrepancy sequence-based initialization approaches using population-based algorithms for solving the global optimization problems," *Applied Sciences*, vol. 11, no. 16, pp. 7591, 2021.

[6] S. Pervaiz, Z. Ul-Qayyum, W. H. Bangyal, L. Gao and J. Ahmad, "A systematic literature review on particle swarm optimization techniques for medical diseases detection," *Computational and Mathematical Methods in Medicine*, vol. 2021, no. 5, pp. 1–10, 2021.

[7] W. Qi, "Object detection in high resolution optical image based on deep learning technique," *Natural Hazards Research*, pp. S2666592122000531, 2022. https://doi.org/10.1016/j.nhres.2022.10.002

[8] F. N. Venkateswaran and K. Umadevi, "Hybridized wrapper filter using deep neural network for intrusion detection," *Computer Systems Science and Engineering*, vol. 42, no. 1, pp. 1–14, 2022.

[9] N. Dilshad, A. Ullah, J. Kim and J. Seo, "LocateUAV: Unmanned aerial vehicle location estimation via contextual analysis in an IoT environment," *Internet of Things Journal*, pp. 1, 2022. https://doi.org/10.1109/JIOT.2022.3162300

[10] N. Dilshad and J. Song, "Dual-Stream siamese network for vehicle re-identification via dilated convolutional layers," in *IEEE Int. Conf. on Smart Internet of Things*, Jeju Island, SK, pp. 350–352, 2021.

[11] S. Malek, F. Melgani, M. Mekhalfi and Y. Bazi, "Real-time indoor scene description for the visually impaired using autoencoder fusion strategies with visible cameras," *Sensors*, vol. 17, no. 11, pp. 2641, 2017.

[12] S. Ren, K. He, R. Girshick, X. Zhang and J. Sun, "Object detection networks on convolutional feature maps," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 7, pp. 1476–1481, 2017.

[13] Y. LeCun, Y. Bengio and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[14] K. Chatfield, K. Simonyan, A. Vedaldi and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *Proc. of the British Machine Vision Conf. 2014*, Nottingham, pp. 6.1–6.12, 2014.

[15] S. Y. Rhyou, H. J. Kim and K. Cha K, "Development of access management system based on face recognition using ResNet," *Journal of Korea Multimedia Society*, vol. 22, no. 8, pp. 823–831, 2019.

[16] Y. Zahid, M. A. Tahir and M. N. Durrani, "Ensemble learning using bagging and inception-V3 for anomaly detection in surveillance videos," in *2020 IEEE Int. Conf. on Image Processing (ICIP)*, Abu Dhabi, United Arab Emirates, pp. 588–592, 2020.

[17] J. Li, W. Chen, Y. Sun, Y. Li and Z. Peng, "Object detection based on DenseNet and RPN," in *2019 Chinese Control Conf. (CCC)*, Guangzhou, China, pp. 8410–8415, 2019.

[18] H. Alhichri, Y. Bazi, N. Alajlan and B. Bin Jdira, "Helping the visually impaired see via image multi-labeling based on squeezenet CNN," *Applied Sciences*, vol. 9, no. 21, pp. 4656, 2019.

[19] A. Kaveh and A. Dadras, "A novel meta-heuristic optimization algorithm: Thermal exchange optimization," *Advances in Engineering Software*, vol. 110, no. 4598, pp. 69–84, 2017.

[20] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn and A. Zisserman, "The pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[21] P. Saranya and S. Prabakaran, "Automatic detection of non-proliferative diabetic retinopathy in retinal fundus images using convolution neural network," *Journal of Ambient Intelligence and Humanized Computing*, vol. 1, no. 2, pp. 9, 2020.

[22] S. Zhang, F. Song, T. Lei, P. Jiang and G. Liu, "MKLM: A multiknowledge learning module for object detection in remote sensing images," *International Journal of Remote Sensing*, vol. 43, no. 6, pp. 2244–2267, 2022.