A Hybrid WOA-MTCNN Algorithm for Accurate Face Detection Based on IoT Architecture

Xi Hu[1,2,3], Hang Ruan[2,3], Xin Xiong[1,2,3], Siqi Zhang[2,3,4(✉)], Ning Huang[1], and Jun Wang[2,3]

[1] State Key Laboratory of Precision Blasting, Jianghan University, Wuhan 430056, China
[2] School of Artificial Intelligence, Jianghan University, Wuhan 430056, China
sinkia0723@gmail.com
[3] Artificial Intelligence Institute, Jianghan University, Wuhan 430056, China
[4] Dongfeng Motor Corportation Technical Center, Wuhan 430058, China

Abstract. To further improve the detection accuracy in face detection based on IoT architecture, this paper designs and implements a hybrid face detection system based on the MTCNN model, and optimize the MTCNN model by combining Whale Optimization Algorithm (WOA). Firstly, we select and train the loss function weights and batch_size of MTCNN to estimate the model. Secondly, we utilize the WOA algorithm to further optimize the weights and batch_size parameters. Thirdly, we train and simulate the prediction by applying the obtained optimal parameters. Finally, we design and implement a face detection software based on the MTCNN model to test the practical application capabilities of the model.

Keywords: Convolutional Neural Network (CNN) · Deep learning · Face detection · Image pyramid · Whale Optimization Algorithm (WOA)

1 Introduction

With the rapid increase in video and image database, there is an incredible requirement of face detection for automatic target detection of face information based IoT (Internet of Things) architecture, which has a wide range of uses to greatly improve the detection efficiency for facilitating people' lives [1–6]. The IoT have uncommonly extended the proportion of information over the late years, which prepares abundent significant information in real-time applications [2]. Face detection technology as a computer technology applies the deep learning algorithm as its mathematical foundation theory to determine the location, size and color of a human face in a digital image by utilizing varieties of smart electronic products [7]. Convolutional Neural Network (CNN) model is often considered as a relatively basic network model to optimize the deep learning algorithm, which is commonly used in the field of face detection. Therefore, numerical researchers have paid more attention to optimize the methods of face detection for improving the detection accuracy.

X. Chen et al. (Eds.): IoTaaS 2023, LNICST 585, pp. 291–303, 2025.
https://doi.org/10.1007/978-3-031-70507-6_22

Inspired by the idea of cascade classification, early deep learning based models are based on the CNN architecture [9]. With the in-depth research on the CNN architecture, researchers have further proposed more detection models on its basis [1, 9]. The authors in [10] have cascaded the CNN structure on the basis of the waterfall idea of the Viola-Jones algorithm, which can operate at multiple resolutions and has stronger recognition ability than the single network structure. Chen in [11] have proposed Suppervised Transformer Networks (STN) to improve the efficiency of face detection by combining Region Proposal Network (RPN) and R-CNN algorithm. However, how to further improve the detection accuracy is still a hotspot and sticking point for researchers.

Whale Optimization Algorithm (WOA) is a meta-heuristic optimization algorithm proposed by Mirjalili et al. in 2016, which has the advantages of simple mechanism, fast convergence and strong optimization ability [12]. The Whale Optimization Algorithm (WOA) has been widely researched and applied in various fields such as energy management [13], hydroelectric power generation prediction [14, 15], and the improvement of rumor refutation [16], owing to its inherent characteristics since its introduction. Abuntant researchers dedicate their studies to optimize the method of face detection and improve its detection accuray. The authors in [17] have proposed a framework uses data augmentation to refine the original dataset, transfer learning to fine-tune the pre-trained CNN model DenseNet201, and improved WOA for feature selection. The proposed framework achieves high classification accuracy and robustness on benchmark datasets. In [18], the authors have proposed a FER process using WOA-TLBO based MultiSVNN and demonstrate its effectiveness in terms of accuracy. The proposed FER arrangement comprises three phases: feature extraction, feature optimization, and emotion recognition. The authors in [19] have proposed a method for object detection based on the combination of Non-maximum Suppression (NMS) and the chaotic whale optimization algorithm and the proposed method can significantly improve the Average Precision (AP) of detectors compared with the most advanced methods.

However, these existing researches are limited in the following aspects.

i) In current research, CNN has shown stronger accuracy and generalization ability compared to traditional methods for face detection. However, CNN's detection performance is not optimal in the presence of different lighting, angles, facial expressions, or face occlusion.
ii) Parameter settings have a significant impact on the convergence speed and generalization ability of CNN. How to select appropriate parameters is a concern of many researchers.
iii) CNN requires significant computing resources and time to handle large-scale data, which may pose limitations in practical applications. How to improve CNN's ability to process large-scale data is an urgent problem to be solved.

To tackle these limitations, a hybrid WOA algorithm based Multi-Task Convolutional Neural Network (WOA-MTCNN) is proposed in this paper to further improve the detection accuracy for accurate face detection. The major contributions are summarized as follows:

i) We employed a cascaded CNN network structure which has shown to better detect faces in different environments, thus improving detection accuracy and generalization capability.
ii) We utilized an original heuristic optimization algorithm to optimize the parameters of CNN, in order to improve its detection accuracy.
iii) Combine the original heuristic optimization algorithm with MTCNN to optimize the training time of the model and improve its speed in processing large-scale data.

This paper studies the method of face detection, uses a MTCNN to build a detection model, which based on Python + Pytorch + PyQt and other technologies, and uses the WOA algorithm to optimize the model loss function weight and batch_size to improve the model. The purpose of face detection accuracy, and build a three-layer network structure to realize the face detection system.

The rest of this paper is organized as follows. Section 2 introduces the relevant theoretical knowledge of MTCNN. Section 3 introduces the WOA algorithm. Section 4 explains how to construct the WOA-MTCNN algorithm, and Sect. 5 presents the experimental results and practical applications of the algorithm.

2 MTCNN Model

The methodological principle of the MTCNN model can be summarized as: image pyramid + three-stage cascaded CNN. This chapter will briefly introduce the concept of CNN and image pyramid, and then detail the network structure and each loss function of the MTCNN model. The contents are as follows:

2.1 CNN

Convolutional Neural Network (CNN) model is a widely used feedforward neural network model. Inspired by the idea of cascaded classifiers, many early deep learning based models are based on cascaded CNN architectures [8]. In the past, the Full Con-nect Neural Network mainly had three obvious defects in the processing of large-size images, such as spatial information loss, training difficulties, and network overfitting, and the CNN model solved these problems very well [9].

A typical CNN model is mainly composed of the following three parts:

1. Convolutional layer: usually contains multiple learnable convolution kernels, which ex-tract image features by using "filters" (convolution kernels).
2. Pooling layer: reduce the number of parameters in the network and prevent overfitting.
3. Fully connected layer: implement classification and output results.

2.2 Image Pytamid

When discussing image pyramids, an important concept needs to be addressed first: scale. Scale is the size and resolution of an image. Before operating on an image, it is often necessary to pre-process the size of the image to adjust it to the target image needed for the study. And image pyramid refers to an important way of multi-scale image

adjustment expression, which is mainly applied to image segmentation and fu-sion. Each layer of the image pyramid is derived from the same original image, only the scales of different layers are different. The number of generated layers of the image pyramid is mainly determined by two parameters, Minisize, which refers to the mini-mum face size in the input image as considered by the developer, and factor, which is the scaling factor of each layer to the edge length of the previous image. Figure 1 shows a schematic diagram of the image pyramid structure.

Fig. 1. Image pyramid

As from Fig. 1, the image pyramid can be divided into two generation methods by sampling direction.

1. Downward sampling: converting the image from level 0 to level 1, level 2, level 3, the image resolution keeps decreasing and the image becomes smaller.
2. Upward sampling: convert the image from level 3 to level 2, level 1, level 0, the image resolution continues to increase, and the image becomes larger.

In practical applications, the following two image pyramids are often used.

1. Gaussian pyramid: used for down-sampling, widely used in the field of image processing.
2. Laplacian pyramid: used to reconstruct an image by sampling up from the bottom of the pyramid.

2.3 MTCNN Net-Structure

MTCNN is an extension of the model proposed by Shenzhen Research Institute based on Li et al. [10] [16] for the two tasks of joint face detection and face keypoint localization. The model con-sists of Proposal Network (hereafter referred to as “P-Net”), Refine Network (hereafter referred to as “R-Net”), and Output Network (hereafter referred to as "The MTCNN Pipeline is shown in Fig. 2 [16].

P-Net is a fully convolutional neural network, similar to the idea of sliding window, which scans and detects whether each 12*12 region in each layer of the incoming image pyramid contains a face. If the region contains a face, the candidate frame of the face is returned, and after further obtaining the region of the candidate frame corresponding

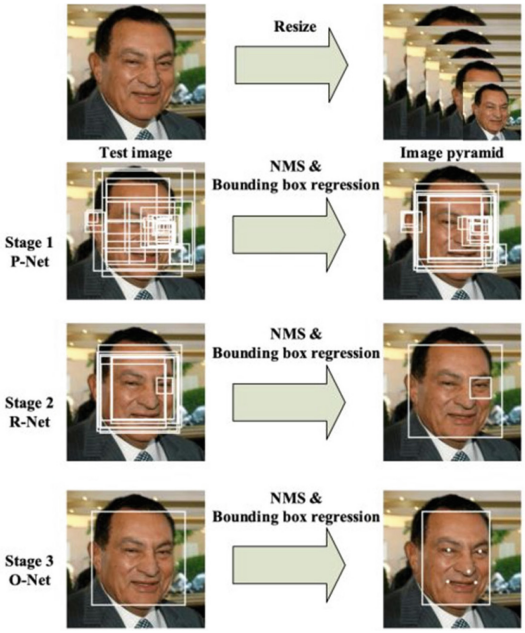


Fig. 2. MTCNN Pipeline

to the original image, the candidate frame with the highest score is retained and the candidate frame with too large overlapping area is removed by NMS.

R-Net is a simple convolutional neural network that bilinearly interpolates the candidate frames that P-Net thinks may contain faces to 24 × 24. In this layer of the network, we will further determine whether they contain faces, and if they do, we will also regress the candidate frames and their corresponding regions of the original image and filter them by NMS.

The O-Net is similar to the R-Net, it is a simple convolutional neural network, and the candidate frames that the R-Net in the previous layer thinks may contain faces are first bilinearly interpolated to 48 × 48 and used as the input of the O-Net. The final discrimination is performed on the image output from R-Net.

2.4 MTCNN Model Parameters

The most important metric for all convolutional neural network training is the loss function. In MTCNN, there are three tasks, which are: face and non-face classification, regression of face bounding box and face key point localization. In the following, the loss functions of the three tasks are explained in this paper.

1. Classification of faces and non-faces. It is actually a binary classification problem for faces, and for the input sample xi, a cross-entropy function is used, as follows.

$$L_i^{det} = -(y_i^{det}\log(pi) + (1 - y_i^{det})(1 - \log(pi)))$$

2. Face bounding box localization. For face target box regression, Euclidean distance is taken as.

$$L_i^{box} = \left|\left|\hat{y}_i^{box} - y_i^{box}\right|\right|$$

where $\hat{y}_i^{box}$ represents the coordinates of the bounding box corrected after the grid output, and y_i^{box} represents the real bounding box of the target face.

3. Face key point localization. Again Euclidean distance is taken to calculate.

$$L_i^{landmark} = \left|\left|\hat{y}_i^{landmark} - y_i^{landmark}\right|\right|$$

The $\hat{y}_i^{landmark}$ represents the keypoint coordinates obtained after the network calculation, and $y_i^{landmark}$ is the real coordinates of keypoints, which contains the horizontal and vertical coordinates of five keypoints. In this paper, we only focus on face detection, so keypoint localization is only used for introduction.

Combining the above three loss functions according to the different weights set, the total loss is:

$$min \sum_{i=1}^{N} \sum_{j \in \{det, box, landmark\}} \alpha_j \beta_i^j L_i^j$$

It should be noted that due to the differences in specific tasks of each layer of the net-work, the weights of each loss function are different for each layer of the network, and how to choose the appropriate weights for training to achieve the optimal training effect is a question worth exploring. In this paper, the joint whale optimization algorithm is chosen to optimize in order to find the optimal weight values.

Intersection-over-Union (IOU) is also a very important metric in model training. IOU is used to calculate the overlap of two overlapping images. The higher the overlap, the larger the value of the IOU. As shown in Fig. 3, the IOU is mainly used in applications related to target detec-tion. In this paper, a model is trained to output a regression frame that fits perfectly around an object. For example, in the image below, there is a green regression box and a blue regression box. The green box represents the true correct regression box and the blue box represents the regression box predicted by the model proposed in this paper. The goal of this model is to continuously improve its predicted values until it reaches the optimal situation where the blue box overlaps the green box completely, i.e., the IOU between the two boxes is equal to 1.

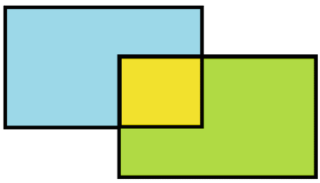


Fig. 3. IOU

The setting ratio for the model IOU is shown in Table 1. The three-layer model in the original MTCNN can achieve 94.6% accuracy in P-Net, 95.4% accuracy in R-Net, and 95.4% accuracy in O-Net.

Table 1. (IOU threshold definition).

Data Type	Negative	Positive	Part
IOU	< 0.3	> 0.65	0.4 ~ 0.65

3 Whale Optimization Algorithm

Whale Optimization Algorithm (WOA) is a meta-heuristic optimization algorithm proposed by Prof. Mirijalili in 2016 [7]. The algorithm seeks the optimal objective by simu-lating the behavioral pattern of humpback whale hunting.

3.1 Algorithm Fundamentals

Humpback whales have a special hunting method called bubble net foraging. In nature, humpback whales usually live in groups. During the feeding process, humpback whales will surround their prey in groups, spitting out bubbles during the spiral movement, forming a spiral "bubble net" and then forcing the prey tighter and tighter. WOA simulates this special feeding mechanism of humpback whales, which includes: encircling the prey, bubble net feeding mode (local search), searching for prey (global search).

4 Build WOA-MTCNN Algorithm

In this paper, the following comparisons and improvements are made to address the charac-teristics of the MTCNN model.

1. Minisize is dynamically set for the input image size to be detected to better adapt to the faces in different size images, so that the number of image pyramid layers to be de-tected by the incoming model is reasonable and the detection rate is improved.
2. The effects of different weight loss functions on the accuracy and recall of de-tection for the same batch_size in each layer of the network are compared.
3. Compare for different batch_size and study and analyze its effect on gradient descent direction.
4. Combine the whale optimization algorithm to find out the optimal loss function weights and batch_size.

The flow chart of the algorithm is as follows:

Step 1. Initialize the loss function weights and batch_size of MTCNN.

Step 2. Use the WOA algorithm to continuously update the individual whale positions until the abort criterion is satisfied.

Step 3. Output the optimal weights and batch_size parameters.

Step 4. Obtain the optimal parameters, train and simulate the prediction.

The specific flow chart is shown in Fig. 4.

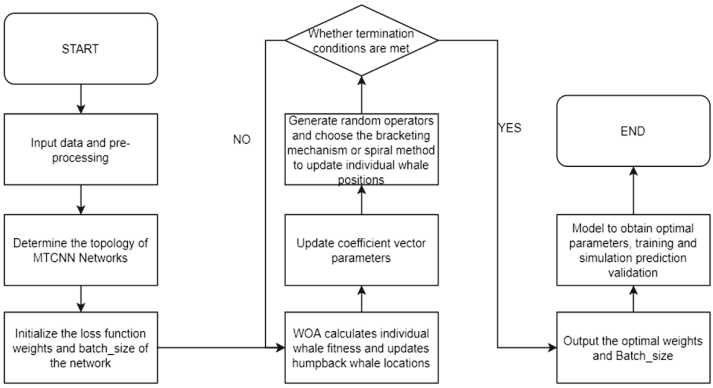


Fig. 4. Flow chart of the MTCNN algorithm

4.1 Experimental Design

In this paper, three sets of experiments are designed to compare and analyze the model accuracy and improve it.

Experiment 1.

Take the same loss function weights, modify the batch_size only, and compare the effect of different batch_size of 256 and 512 groups on the direction of gradient descent of the model.

Experiment 2.

Adopt the same batch _size, modify the loss function weights of the three-layer network, and compare the effect of different weights of each group on the accuracy of the model.

5 Evaluation Criteria

After training by modifying the individual variables, the accuracy of each layer of the network is compared to find the optimal combination for improving the accuracy of the model.

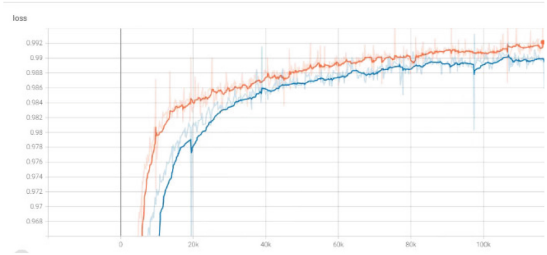
5.1 Experiment Result and Analysis

After training by modifying the individual variables, the accuracy of each layer of the network is compared to find the optimal combination for improving the accuracy of the model.

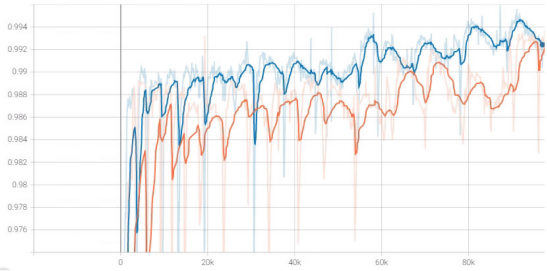
Experiment 1 results analysis

The loss function weights of 1:1 face classification and face frame regression are uniformly selected for this set of experimental three-layer network, and the batch_size is divided into two groups of 256 and 512 for training comparison analysis.

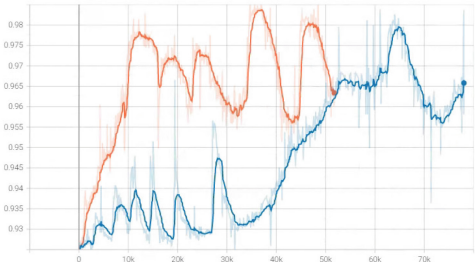
The experimental results are shown in Fig. 5 (orange Batch_size = 512, blue Batch_size = 256).



(a). cls_weight = offset_weight P-net Accuracy



(b). cls_weight = offset_weight R-net Accuracy



(C). cls_weight = offset_weight O-net Accuracy

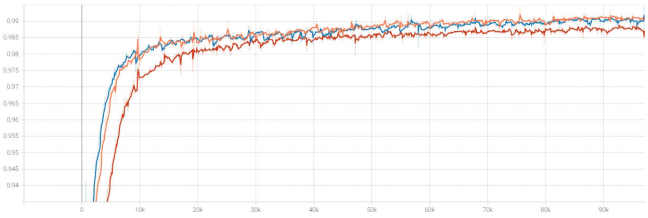
Fig. 5. Different batch_size, same cls_weight

From Fig. 5(a), it can be concluded that in the P-layer network, both sets of data start to converge at 20k samples, and the accuracy increases with the increase of training data, and the training effect of Batch_size = 512 sets is better than that of Batch_size = 256 sets. However, the training effect is reversed for R-layer and O-layer networks, and the effect is better for Batch_size = 256 groups. Since the O-layer net-work is the final processing part of the whole network structure, which will get the final output results, and the accuracy rate of Batch_size = 512 groups fluctuates significantly by two factors, this paper believes that Batch_size = 256 can bring higher accuracy rate.

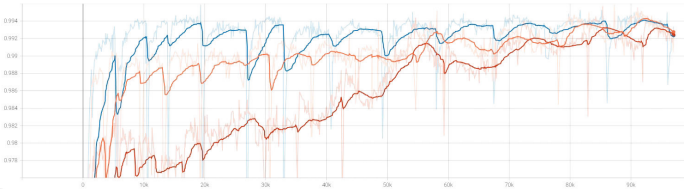
Experiment 2 Result Analysis

This group of experiments was uniformly selected with batch_size = 512, and the loss function of the three-layer network was divided into three groups for comparison training.

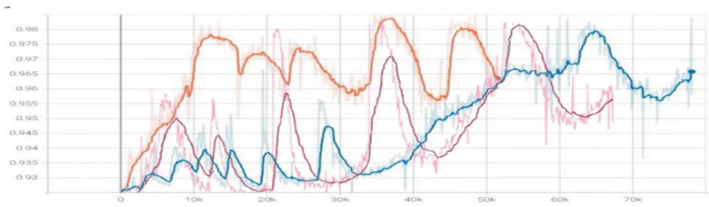
The experimental results are shown in Fig. 6 below (red cls_weight = 0.2, orange cls_weight = 0.5, blue cls_weight = 0.8).



(a). batch_size = 512 P-net accuracy



(b). batch_size = 512 R-net weight accuracy



(c) batch_size = 512 O_net weight accuracy

Fig. 6. Same batch_size, different cls_weight

From Fig. 6(a), it can be concluded that all three groups converge well in the P-layer network, and the training effect is relatively poor for cls_weigh = 0.2. In the P-layer and O-layer networks, all three groups have large fluctuations, and cls_weight = 0.8 can converge faster and achieve higher accuracy in comparison. Therefore, the weight of classification loss function (cls) should be higher than the weight of regression loss function (offset) in order to obtain better training results.

5.2 Build and Demonstrate of the System

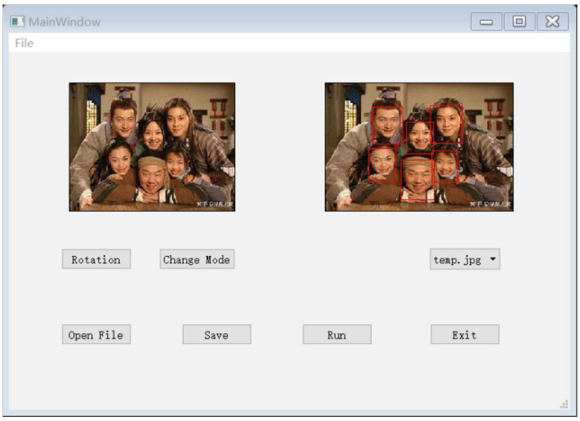
Our face detection system GUI interactive interface is used in Python PyQt5 framework to design and build, using Pyinstaller to package the system as a desktop application.

The main functions of the in-person face detection system are.

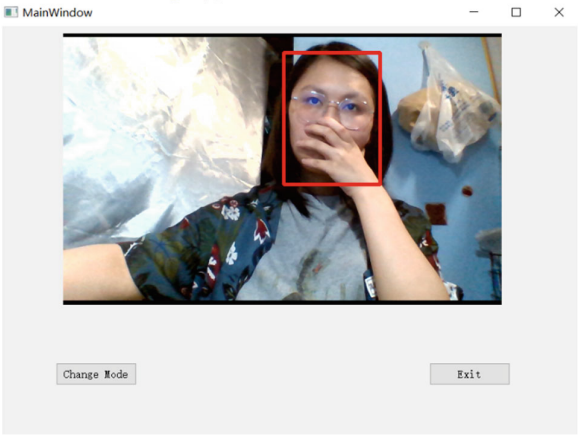
1. image uploading and rotating and saving.
2. detection of faces in the input static images.
3. Real-time detection of faces in the video stream captured by the camera.

The upper part of the software is the image box, and the lower part is the various function buttons. The left side of the upper half is the input box, the selected image will be displayed in the left image box, and the completed detected image will be displayed in the right output box. The buttons in the lower half are used to implement various functions. You can select the local image by Open File button or Ctrl + N, run the program by Run button, call the model to detect the image, then save the image with the face labeled after detection to local by Save File button or Ctrl + S, and finally exit the program by Exit button or Esc. If the direction of the input image is not positive, the image can be rotated by the Rotation button. If real-time face detection is required, you can switch the mode to real-time video detection mode by clicking the Change Mode button. You can switch between two modes of static image face detection and real-time face detection at any time by clicking the Change Mode button.

From Fig. 7, it can be seen that the system is able to adapt to different sizes of static and dynamic face detection, and can still accurately detect the face area even when part of the face is occluded.



(a). Static face detection



(b). Real-time face detection

Fig. 7. Face detection system GUI

Acknowledgment. This work was supported in part by the National Natural Science Foundation of China under Grant 61901298, 71601085; in part by the Key Research and Development Project of Hubei Province, China under Grant 2020BCA084; in part by the Scientific Research Project of Education Department of Hubei Province under Grant B2022280; in part by the Young Talents Science and Technology Innovation Planning Program of Education Department of Hubei Province under Grant T2022045; in part by the Scientific Research Foundation of Jianghan University under Grant 2023KJZX18.

References

1. Cao, C., Cao, Z., Cui, Z.: LDGAN: a synthetic aperture radar image generation method for automatic target recognition. IEEE Trans. Geosci. Remote Sens. **58**(5), 3495–3508 (2020)
2. Chauhan, D., Kumar, A., Bedi, P., Athavale, V.A., Veeraiah, D., Pratap, B.R.: An effective face recognition system based on Cloud based IoT with a deep learning model. Microprocess. Microsyst. **81**, 103726 (2021)

3. Hu, P., Ning, H., Qiu, T., et al.: Security and privacy preservation scheme of face identification and resolution framework using fog computing in internet of things. IEEE Int. Things J. **4**(5), 1143–1155 (2017)
4. Firouzi, F., Farahani, B., Barzegari, M., Daneshmand, M.: AI-driven data monetization: the other face of data in IoT-based smart and connected health. IEEE Int. Things J. **9**(8), 5581–5599 (2020)
5. Aydin, I., Othman, N. A.: A new IoT combined face detection of people by using computer vision for security application. In 2017 IEEE International Artificial Intelligence and Data Processing Symposium (IDAP) pp. 1–6. (2017)
6. Chen D., G. Hua, F. Wen, and J. Sun, Supervised transformer network for efficient face detection. in European Conference on Computer Vision. Springer, pp. 122–138 (2016)
7. Kumar, A., Kaur, A., Kumar, M.: Face detection techniques: a review. Artif. Intell. Rev. **52**, 927–948 (2019)
8. Minaee, Shervin, et al. Going deeper into face detection: a survey. arXiv preprint arXiv:2103.14983 (2021)
9. Alzubaidi, L., Zhang, J., Humaidi, A.J., et al.: Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. J. Big Data **8**, 1–74 (2021)
10. Li, H., Lin, Z., Shen, X., Brandt, J., Hua, G.: A convolutional neural network cascade for face detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp.5325–5334 (2015)
11. Chen, D., Hua, G., Wen, F., et al.: Supervised transformer network for efficient face detection. Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14. Springer International Publishing, pp. 122–138 (2016)
12. Mirjalili, S., Lewis, A.: The whale optimization algorithm. Adv. Eng. Softw. **95**, 51–67 (2016)
13. Xiong, X., Hu, X., Guo, H.: A hybrid optimized grey seasonal variation index model improved by whale optimization algorithm for forecasting the residential electricity consumption. Energy **234**, 121127 (2021)
14. Xiong, X., Hu, X., Tian, T., Guo, H., Liao, H.: A novel Optimized initial condition and Seasonal division based grey seasonal variation index model for hydropower generation. Appl. Energy **328**, 120180 (2022)
15. Li, Z.K., Hu, X., Guo, H., Xiong, X.: A novel weighted average weakening buffer operator based fractional order accumulation seasonal grouping grey model for predicting the hydropower generation. Energy **277**, 127568 (2023)
16. Hu, X., Xiong, X., Wu, Y., Shi, M.J., Wei, P., Ma, C.M.: A hybrid clustered SFLA-PSO algorithm for optimizing the timely and real-time rumor refutations in online social networks. Expert Syst. Appl. **212**, 118638 (2023)
17. Hussain, N., Khan, M.A., Kadry, S., et al.: Intelligent deep learning and improved whale optimization algorithm based framework for object recognition. Hum. Cent. Comput. Inf. Sci **11**(34), 1–17 (2021)
18. Lakshmi, A.V., Mohanaiah, P.: WOA-TLBO: Whale optimization algorithm with Teaching-learning-based optimization for global optimization and facial emotion recognition. Appl. Soft Comput. **110**, 107623 (2021)
19. Wu, G., Li, Y.: Non-maximum suppression for object detection based on the chaotic whale optimization algorithm. J. Vis. Commun. Image Represent. **74**, 102985 (2021)
20. Muthaiah, U., Chitra, S.: Mango pest detection using entropy-ELM with whale optimization algorithm. Intell. Autom. & Soft Comput. **35**(3) (2023)