Hindawi
Journal of Electrical and Computer Engineering
Volume 2024, Article ID 3443028, 21 pages
https://doi.org/10.1155/2024/3443028

Research Article

A Novel Technique for Facial Recognition Based on the GSO-CNN Deep Learning Algorithm

Rana H. Al-Abboodi [1] **and Ayad A. Al-Ani** [2]

[1]*Information Engineering College, Al-Nahrain University, Baghdad, Iraq*
[2]*Department Name of Organization, Al-Nahrain University, Baghdad, Iraq*

Correspondence should be addressed to Rana H. Al-Abboodi; rana.alabboodi.st2022@nahrainuniv.edu.iq

Received 22 September 2023; Revised 24 January 2024; Accepted 16 April 2024; Published 20 May 2024

Academic Editor: Gongping Yang

Copyright © 2024 Rana H. Al-Abboodi and Ayad A. Al-Ani. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Face recognition is one of the important elements that can be used for securing the facilities, emotion recognition, sentiment exploration, fraud analysis, and traffic pattern analysis. Intelligent face recognition has yielded excellent accuracy in a controlled environment whereas vice versa in an uncontrolled environment. However, conventional methods can no longer satisfy the demand at present due to their low recognition accuracy and restrictions on many occasions. This study proposed an optimal deep learning-based face recognition system that improves the security of the model developed in the IoT cloud environment. Initially, the dataset of images was gathered from the public repository. The captured images are explored using image processing techniques like image preprocessing employing the Gaussian filter technique for removing the noise and smoothing the image. The histogram of oriented gradients (HOGs) is used for the image segmentation. The processed images are preserved at the cloud service layer. Extract features were linked to facial activities using the spatial-temporal interest point (STIP). On the other hand, the extracted feature vectors are investigated using galactic swarm optimization (GSO) techniques that give optimized feature vectors. The necessary features are selected using the gray level co-occurrence matrix (GLCM), which separates the statistical texture features. The GSO output is fed into the deep convolutional neural network (DCNN) that effectively trains the captured face images. This will allow the effectiveness of the GSO-CNN technique to be assessed in terms of recognition accuracy, recall, precision, and error rate.

1. Introduction

The Internet of Things (IoTs) refers to a network of computing devices, cars, and buildings that can collect and share data via the Internet using technologies such as embedded software and sensors. IoT in its simplest form enables items to communicate with one another and with centralized systems or platforms, creating a web of interconnected gadgets that can offer helpful insights and automation. In addition, IoT devices are completely connected through a device controller known as a smartphone [1]. The impact of the Internet of Things on people's commercial, economic, and social lives has grown significantly in recent years [2]. IoT is the process of associating embedded wired and wireless technologies with the help of distributed and interlinked network structures. The widespread availability of IoT devices in the current world has greatly improved people's quality of life. IoT networks are susceptible to cyberattacks because they are inherently unstable, unstructured, and under-resourced. Maintaining constant monitoring and analysis of IoT systems allows for the adoption of preventative actions, the mitigation of risks, and the safeguarding of sensitive data.

IoT security is considered to be a significant area of research nowadays. As the modern era is technology dominated, communication is achieved using modern technologies; for instance, IoT devices [3] are responsible for providing various services including communication and hence it is required to undergo authentication procedures in

a secured manner to ensure trustworthy and secure communication [4]. However, maintaining IoT security is challenging due to its diversified nature.

Real-time applications in various domains, including security and disease detection, have benefited greatly from the recent advancements achieved in image processing techniques (IPTs) [5, 6]. Image categorization, concept understanding, and location computation are all areas that must be prioritized to get a well-rounded understanding of the acquired images. The term "object detection" [7] describes this practice. In addition to facial recognition, skeletal detection, pedestrian detection, text recognition, and so on are also supported. According to the findings, object detection methods are used to solve the vast majority of computer vision issues. Images and videos are analyzed to extract the most relevant data, which is then used to solve problems in areas such as image classification, human behavior analysis, and intelligent driving [8, 9].

Security measures must protect data from change and scrutiny to maintain privacy, dependability, and accessibility [10]. Successful IoT face recognition research has led to many applications [11, 12]. Deep learning uses advanced neural networks. Multilayered feature extracted to enhance classification is input data. Face recognition experiments recognize faces from images [13]. This is because typical data processing technology cannot handle enormous amounts of data. The data analysis technique [14] and its frameworks for storing and processing huge data assist in overcoming the problems of analyzing diverse and complicated data to obtain business-relevant information [15].

Different techniques have been utilized in this study to capture crucial facial features for recognition. Deep learning algorithms are frequently used with existing techniques. In face recognition tasks, GSO-CNNs can be efficiently developed while still maintaining high accuracy. Compact models that are appropriate for IoT devices can be made using methods such as quantitative analysis, elimination, and data extraction. The shortcomings of existing techniques are addressed, and a useful face recognition system for the Internet of Things is also addressed. As a result, we proposed the GSO-CNN for face recognition in an IoT environment.

1.1. Motivation and Objectives. The study aims to design an optimal and enhanced security model for the IoT cloud environment for face recognition systems using a deep learning approach. Classifiers and feature vectors worked well in controlled environments in previous experiments. Most surveillance-based application systems use biometric authentication modules to secure smart environments. The frontal face has a neutral look and less illumination with varied stances. Thus, face photos are being employed to improve IoT security.

(i) Poor facial recognition: previous investigations have shown that face photos do not operate adequately on model components with locally aggregated descriptors. Thus, gender and facial emotion recognition will not improve. Deep learning concepts have been introduced to have insights into face images with lowered illumination; however, the performance of the IoT environment needs to be addressed.

(ii) Lack of system: the essential infrastructure of IoT is combined with cloud-based services that facilitate data storage, data processing, and data sharing. Data storage and transmission nodes and computer devices in the Internet of Things are frequent targets of hackers and other malicious actors. Security issues and opportunities to implement and enforce privacy and security safeguards emerge at each successive level of the IoT architecture.

(iii) Lack of security: the safety of the system and its ability to conserve energy require an analysis of certain security measures. To ensure the safety of the network, the cloud layer designs a protocol for communication among edge nodes, fog nodes, and sensors. No amount of message-passing protocol, point-to-point encryption, or doing away with certificates has reduced the amount of data snooping and logging that goes on. However, the data fusion module allows hackers to impersonate legitimate users and access sensitive information. Due to their transient nature, Internet of Things devices introduce novelty flaws into an already vulnerable network of sensors and data sources. Therefore, new intelligent and adaptable security solutions are needed.

The main objective of the research is a face recognition system that improves the security of the model developed in the IoT cloud environment. The following are some of the research objectives:

(i) To enhance DNN with the aid of a hybrid optimization algorithm.

(ii) To improve the efficiency of the segmentation approach.

(iii) To extract the relevant facial features from the image frames for designing the security.

(iv) To develop an intelligent object detection model in terms of accuracy.

1.2. Research Contributions. The research goal was to face recognition system that improves the security of the model developed in the IoT cloud environment. The primary contribution of this work is as follows:

(i) The image was collected from the public repository.

(ii) Image preprocessing using a Gaussian filter for removing the noise and smoothing the image.

(iii) Image segmentation is done by histogram orientation gradients for object detection of the image process.

(iv) The GLCM's feature selection can be utilized to evaluate and quantify the texture characteristics in these images.

(v) Spatial-temporal interest point (STIP) is employed to extract the features related to facial behaviors from facial action units (FAUs).

(vi) An improved face recognition approach employing galactic swarm optimized convolution neural network (GSO-CNN) algorithms that provide optimized feature vectors.

(vii) Explore the security significance of cloud computing.

1.3. Paper Organizations. In the rest of the paper, in Section 2, we provide a summary of the relevant literature. Section 3 identifies the precise issue and gives research solutions in the case of comparative works. We detail the planned work and it is necessary; the methodology and model are described in depth in Section 4. Section 5 presents the implementation findings and discusses the operational model. Finally, Section 6, the report, concludes with several suggestions for further research.

2. Literature Survey

Furthermore, this part describes the research gaps of earlier efforts, which are detailed in the following. Cloud-based applications are offered as an illustration model in [16], with layers for infrastructure provisioning, clustered elastic platforms, service composition, and applications. This application stack has been useful for some purposes; however, it does not provide a very high quality of service. Composition in the form of a named, acyclic graph using OSGi services was first presented in a similar work [17]. Further development of this idea led to the development of the hierarchically distributed data matrix (HDM) in [18] to represent big data applications. Also included are various runtime frameworks for Internet of Things restraints. However, large-scale embedded systems could not function properly in cloud stack management. A distributed and declarative cloud-automated architecture called Cloud Orchestration Policy Engine (COPE) [19] was developed to alleviate the negative effects of poor performance on operational needs and service level agreements. For complicated applications that benefit from declaratively defined workflow topologies, the authors of [20] developed a declarative service workflow architecture based on iPOJO. The authors of [21] proposed to have developed a system to identify network intrusions through deep learning. It turns out that the model does better than expected on the NSL-KDD dataset. Because of the proliferation of IoT devices and the ready availability of real-time data, there is a massive delay problem. The research goal of [22] was to create a smart home system with offline/online attendance tracking and offender identification via an image recognition algorithm. Humans have an extremely difficult time remembering faces but computers do not have these problems, thus they can be employed in situations when more photographs need to be stored in facial database entries.

To improve infrared picture production, texture detail, and model stability, researchers in [23] built cycle generative adversarial networks using gradient normalization. To begin, the UNet-generating network's blurring and degrading feature extraction was addressed by using the residual network, which has a higher capacity for feature extraction and hence produces more distinct IR images. In addition, ResNet used channel attention and spatial attention algorithms to weight image features, improving feature perception in salient locations and creating picture details, to compensate for the significant lack of detail in generated infrared images.

The authors of [24] proposed a deep learning-based NID method that makes use of a convolutional bidirectional long short-term memory neural network trained with log-cosh conditional variational auto encoders (LCVAEs). It can build virtual samples with the correct labels from observed traffic data and extract more crucial attack parts. A log-cosh-reconstructed loss term is added to the conditional auto encoder. Virtual samples can inherit discrete attack data from it and improve unbalanced attack features. The spatial and temporal characteristics of the attack are subsequently taken into account using a CNN-BilSTTM hybrid feature extraction model. The tests that follow put the suggested method through its paces on NSL-KDD.

The authors of [25] introduced a revolutionary Internet of Things (IoT)-based face mask detection system for use on buses. Using facial recognition, this technology would gather information in real time. The primary goal of this study is to use deep learning, machine learning, and image processing methods to identify instances of face masks in real-time video streams. A model that combines deep learning with traditional machine learning was developed for this purpose. A fresh dataset, in addition to existing available datasets, was used to test the model. The authors of [26] offered a revolutionary image recognition and navigation system that communicates with visually impaired users via audio, providing them with clear and timely directions. Using ROC analysis, we evaluate how well the suggested strategy performs in comparison to other methods. Major challenges for the visually impaired include overcoming obstacles in both indoor and outdoor settings and identifying the person in front of them. It is challenging to identify things or people using solely perceptual and auditory information. The authors of [27] looked into the use of deep learning models for item recognition and anomaly detection in IoT-enabled smart homes. Excellent performance in detecting differences and facial recognition meant that the models might be used to enhance existing IoT devices for the house. This study demonstrates that smart homes could be made more secure and private by utilizing deep learning techniques. There is a pressing need for additional research into questions like how to measure model generalization, how to create cutting-edge methods like transfer learning and hybrid approaches, how to probe privacy-preserving procedures, and how to address deployment issues.

The authors of [28] proposed a deep model built on trees for use in cloud-based automatic facial recognition systems. The suggested deep model reduces computing overhead while maintaining or improving accuracy. The model works by splitting up a single input volume into several smaller volumes, each of which has its tree constructed. A tree's

distinguishing features include its height and branching ratio. A convolutional layer, batch normalization, and a nonlinear function make up the residual function that stands for each separate branch. Several public datasets are used to test the suggested model. The results are also compared to the best current deep models can offer in facial recognition. The authors of [29] presented a two-layer convolutional neural network (CNN) for learning high-level characteristics for face identification. To put pattern recognition and classification to use in the real world, feature extraction is essential. The accuracy of facial recognition systems can be greatly enhanced by providing a detailed description of the input face image. The popular face classifier known as the sparse representation classifier (SRC) creates a representation of the face image by using only a tiny subset of the training data.

In the study of [30], the authors introduced Faster R-CNN, a unique deep learning-based CNN that integrates with IoT to solve common workplace security problems. The neural network was trained using a library of preprocessed images of currently employed individuals. Quicker R-CNN quickly extracts features from preprocessed images by using VGG-16 as its underlying architecture. The deployment of a deep neural network to tackle the challenges of facial recognition has become viable because of recent advancements in IoT and deep learning. Using images of real-world objects labelled with two labels (with mask and without mask), and the authors of [31] generated a new publicly available dataset named RILFD. In addition to testing YOLOv3 and Faster R-CNN, two commercial deep learning models, several machine learning models are analyzed for their ability to recognize face masks. They suggest combining four steps of image processing alongside targeted convolutional neural network models to identify faces hidden by masks. The computational burden of current convolutional network models when deployed in the IoT context motivated the authors of [32] to propose a lightweight model image-encoded HHAR and they dubbed multiscale image-encoded HHAR (MS-IE-HHAR). The model first employs an improved spatial-wise and Cchannel-wise attention (ISCA) module and then a hierarchical multiscale extraction (HME) module to extract information at various scales. Deep learning architectures enable higher performance with large datasets [33]. The CNN model-based approach addresses this issue. This research employs the CK+ and FER-2013 datasets for facial expression recognition. In this research [34], they use CNN-based deep learning to execute this task. Deep learning does better than machine learning in analyzing unstructured data, movies, and other media. They developed a real-time system that can recognize faces, analyze moods, and propose music. To enhance picture classification performance, the research in [35] utilized Inception-v3, a well-known deep convolutional neural network, with additional deep properties, detecting and classifying emotions using CNN-based Inception-v3. The datasets CK+, FER2013, and JAFFE are used. The recommended model outperforms other machine learning methods. Transfer learning algorithms enable successful image-based sentiment analysis. This research examines alternative transfer learning methods for picture sentiment categorization. We compared the results using popular picture sentiment datasets such as CK+, FER2013, and JAFFE [36]. Deep features and Inception-v3, a famous deep convolutional neural network, increase picture classification in this study [37]. A convolutional neural network based on Inception-v3 architecture classifies emotions using CK+, FER2013, and JAFFE datasets. Table 1 shows the summary of existing research gaps.

3. Major Problem Statement

This section provides a brief overview of the explicit works that are already in existence and their related solutions. The research presented in this study also provides solutions to the problems that are specifically mentioned.

3.1. Issues and Particular Research Works. The authors of [40] proposed the extract of complete and reliable local areas by intensively sampling and sparsely detecting facial points. Convolutional neural networks (CNNs) are then used to create convolution features that are both region aware and identity distinguishing for faces. Together, this detailed facial description and a generic face dataset containing commonly seen facial variants form a joint and collaborative representation framework for capitalizing on the unique and universal qualities shared by different geographic areas.

(i) This framework creates a local representation of the query face image under the condition that all parts of the image have the same representation coefficients.

(ii) A joint and collaborative representation with local adaptive convolution feature (JCR-ACF) is proposed, which fully exploits both discriminative local facial features that are robust to different facial variations and powerful representation dictionaries of facial variations, thereby solving the small-sample-size problem.

The authors of [41] described the face biometric quality assessment (BQA) used in face recognition. The dataset was gathered from the CASIA-web face, LFW, and YouTube Face databases. The BQA model is made more robust against noisy labels by using a lightweight convolutional neural network (CNN) with Max-Feature-Map units. The outcomes show that the suggested BQA procedure is effective. The classification of BQA, particularly light convolutional neural networks (CNNs), has proven effective in overcoming image processing issues.

Convolutional neural networks (CNNs) were proposed by the authors of [42] for 2D and 3D face recognition. Two convolutional neural network (CNN) methods, CNN-1 and CNN-2, are used to evaluate the classification strategies' performance in 2D and 3D face recognition. FRGCv2.0 and AT&T dataset images used two distinct CNN models were then put to the test for 2D and 3D face recognition using raw image input and LBP-processed features.

Table 1: Summary of the literature survey.

References	Aim	Algorithm/deep learning methods/techniques	Limitation
[16]	To examine the background and development of cloud application architectures	Cloud-native applications (CNAs) Service-oriented architectures (SOAs)	The work performed microservices and server less to develop the architecture. This limited service application development and raised user complexity
[17]	Using a directed acyclic graph-like composition model, this study suggests an approach to OSGi service composition	OSGi, directed acyclic graph	Lack of composition support limits service reuse, which is essential for scalability and productivity
[18]	The hierarchically distributed data matrix (HDM) is a strongly typed data format that can be used to build scalable, distributed applications	Hierarchically distributed data matrix	Even though this work performs directed acyclic graphs for big data processing, however, the HDM was constrained by the complexity of the outcome
[19]	To propose a Cloud Orchestration Policy Engine (COPE), an open framework that enables cloud providers to manage cloud resources in a declarative, automated fashion	Cloud Orchestration Policy Engine (COPE)	A particular difficulty is orchestrating complex processes
[20]	To proposed iPOJO flow component-based service workflow architecture makes it simple to assemble various services to create complex real-world applications	iPOJO flow, OSGi	The iPOJO cannot support complex interaction patterns. Its composition support is lacking, limiting its cloud computing leadership
[21]	Throughout this work, an NIDS is developed via a deep-learning-influenced methodology	IoT, fog, deep learning	To address IoT security concerns, an intrusion detection system (IDS) is a monitoring system used to detect harmful network activity. The security measures are ineffective for low-power, resource-constrained IoT devices
[22]	This work is to develop an intelligent system for keeping track of family members' whereabouts, recording attendance both online and offline, and spotting potential thieves using an image recognition technique	Artificial neural network	Machine learning using ANN is employed to solve complex issues. ANN construction is identifying neurons to respond to a brief problem
[23]	To propose a gradient-normalized cycle generative adversarial network approach to the current problem	Generative adversarial network (GAN)	To address the issue that GAN training models are not stable enough, issues of poor infrared image creation, lack of texture detail, and unstable models
[24]	This study presents a deep learning (DL) approach to network intrusion detection (NID) by fusing with a combination of the techniques	Log-cosh conditional variational auto encoder with a convolutional bidirectional long short-term memory neural network (LCVAE-CBiLSTM)	The proposed work of standard ML-based NID algorithms struggles to detect threats because of the lack of sufficient training data
[25]	To identify masks worn by people in live video, this research utilized deep learning, machine learning, and image processing	Convolution neural network (CNN) Deep neural network (DNN)	The model employs DL, ML, and image processing to reliably identify faces and masks with minimal inference time and memory requirements, all while properly forecasting the complexity of the images
[26]	This study offered a revolutionary picture recognition and navigation system that communicates with visually impaired users via audio, giving them clear directions in a timely fashion	Genetic algorithm RBF kernel	Even though this work performs face recognition and navigation systems using neural learning for smart security in IoT, however, it is less expensive

Table 1: Continued.

References	Aim	Algorithm/deep learning methods/techniques	Limitation
[27]	This research focused on the deep learning models that can be used in IoT gadgets for anomaly detection and facial recognition in smart homes	Logistic regression (LR) Convolutional neural network (CNN) Gradient-boosting classifier	Typical issues include small sample sizes, insufficient facial variety, poor representation of real-world variances, aberrant occurrences, and detection of intrusions
[28]	This research proposed a tree-based deep model for cloud-based face recognition	Deep neural network	Serving many IoT devices at once may demand cloud resources and cause scalability issues, impacting overall efficiency
[29]	To suggest a two-layer convolutional neural network (CNN) for learning high-level features that can be applied sparsely for face recognition	Convolutional neural network (CNN)	This work implemented the high-level feature CNN for facial recognition. However, overfitting occurs when models have too many parameters and not enough training data
[30]	To apply a deep learning approach that makes use of the Internet of Things (IoTs) to address existing office safety issues	Deep learning-based Faster R-CNN	Increasing the complexity
[31]	To take images of real-world objects and classify them with two labels (with mask, without mask) to build a new publicly available dataset named RILFD	YOLOv3 and faster R-CNN	This work is performed by DCNN and image processing. However, the influence of dark and low-light environments is not considered which can affect the robustness of the proposed approach
[32]	To discuss treating the HHAR task as a problem of image classification and encoding sensory heterogeneous HHAR data into a three-channel image representation	Heterogeneous human activity recognition (HHAR)	Scalability can be complex and IoT gadgets with limited power supplies
[33]	To evaluate tests adversarial examples against three anomaly detection models based on deep learning methods	Convolutional neural network (CNN), long short-term memory (LSTM), and deep belief network (DBN)	This work is limited progress towards international security standards. However, the highly complex and huge number of parameters could lead to overfitting
[34]	To compare and contrast three distinct models of convolutional neural networks for use in real-world face expression recognition	Convolutional neural network (CNN)	The work of deep learning is to offer potential remedies for issues such as overfitting and excessive computing complexity

Due to the lack of depth information in 2D images, it can be challenging to tell between faces with similar features, especially when those faces are seen from various perspectives.

Typically, specialized sensors or equipment, such as 3D scanners are needed to collect 3D facial data. This makes the large-scale implementation of 3D facial recognition systems more difficult and costly.

The decision between 2D and 3D face recognition techniques depends on the objectives and limitations of the particular application. Both techniques offer advantages and disadvantages.

Deep coupled ResNet was proposed for low and high resolution by the authors of [43]. The coupled mapping (CM) loss function considers both the similarity and discriminatory power of HR and LR characteristics and is used to improve the model parameters of branch networks. The data were collected from LWF and SCface databases for different resolutions of the probe images. The suggested DCR model outperforms existing methods in experiments using LFW and SCface datasets.

(i) The residual connections are crucial for training extremely deep architectures. ResNet has established itself, showing how residual modules can be used to address the deep network degradation issue.

A new family of residual networks based on the tree architecture was proposed by the authors of [44]. Three distinct tree modules were presented, each with the potential to serve as a drop-in replacement for a single convolutional layer or multiple layers in existing networks. The data were collected from the CIFAR-10 database at the Canadian Institute for Advanced Research. It demonstrates that the proposed networks have a higher information density than many well-known networks.

(i) The tree structure used for convoluted networks with various convoluted levels served as the framework for the deep networks. Although the system now has more parameters and is better at detecting objects, its accuracy has not increased.

3.2. Research Solutions. Our research investigated how image noise reduction and smoothing are commonly achieved through preprocessing methods, while segmentation stands out as a robust feature-based technique for object recognition in image processing. STIP extracts facial behavior features from facial action units. The two main problems that caused the detecting performance to drop were feature selection and image matching. We use deep learning to extract data from images. GSO and DCNN are used to optimize output and reduce classification problems in the classifier. IoT devices have limited memory and computing capability. It can be difficult to tune deep learning models for such devices without losing accuracy.

4. Proposed Method

Initially, load the collected dataset (LFW dataset). Next, perform the preprocessing by using a Gaussian filter to remove the noise and smooth the image preprocessing. Next, load the image segmentation and also perform the histogram of oriented gradients (HOGs) process. Then, perform feature extraction by using spatial-temporal interest point (STIP). Next, perform the feature selection process is by using SURF and GLCM. Next, perform the classification by using deep learning-based galactic swarm optimization (GSO) techniques and detect the facial expression. This study classified deep CNN for face recognition in an IoT environment. Figure 1 depicts the proposed architecture. As listed in the following, numerous steps achieve the proposed work's main goals.

(i) Image preprocessing
(ii) Image segmentation
(iii) Feature extraction
(iv) Feature selection
(v) Face recognition classification

4.1. Image Preprocessing Using the Gaussian Filter. When preprocessing images, the Gaussian filter is applied to either eliminate noise or smooth the image. A kernel can be generated using the Gaussian filter, a nonuniform low-pass filter that is based on the Gaussian equation. To implement convolution using the kernel, a convolution mask is created by multiplying a set of image pixels by their corresponding array pixels. Both the kernel and the coefficient decrease in size; the further the kernel is from its center. Since the Gaussian distribution is nonzero everywhere, an arbitrarily big kernel is required, and so arrays of kernel masks of varying sizes have different numerical patterns. We employ this technique to soften the conspicuous object's edges. We obtain a Gaussian filter for this purpose by utilizing the following function. The Gaussian filter is widely used in image preprocessing and computer vision. Edges and significant features are preserved while noise is reduced and images are smoothed. The Gaussian filter convolves the image using a Gaussian kernel and is based on the Gaussian distribution.

$$H(z, w, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} f^{-\left(w^2/2\sigma^2\right)}, \tag{1}$$

where [45] $H(z, w, \sigma)$ is the Gaussian function's value at these y values.

σ determines how much smoothing is applied by adjusting the Gaussian distribution's spread (its standard deviation). The level of smoothing, as well as visual details such as margins and borders, improves with increasing amounts of σ^2. A picture is filtered using a Gaussian kernel

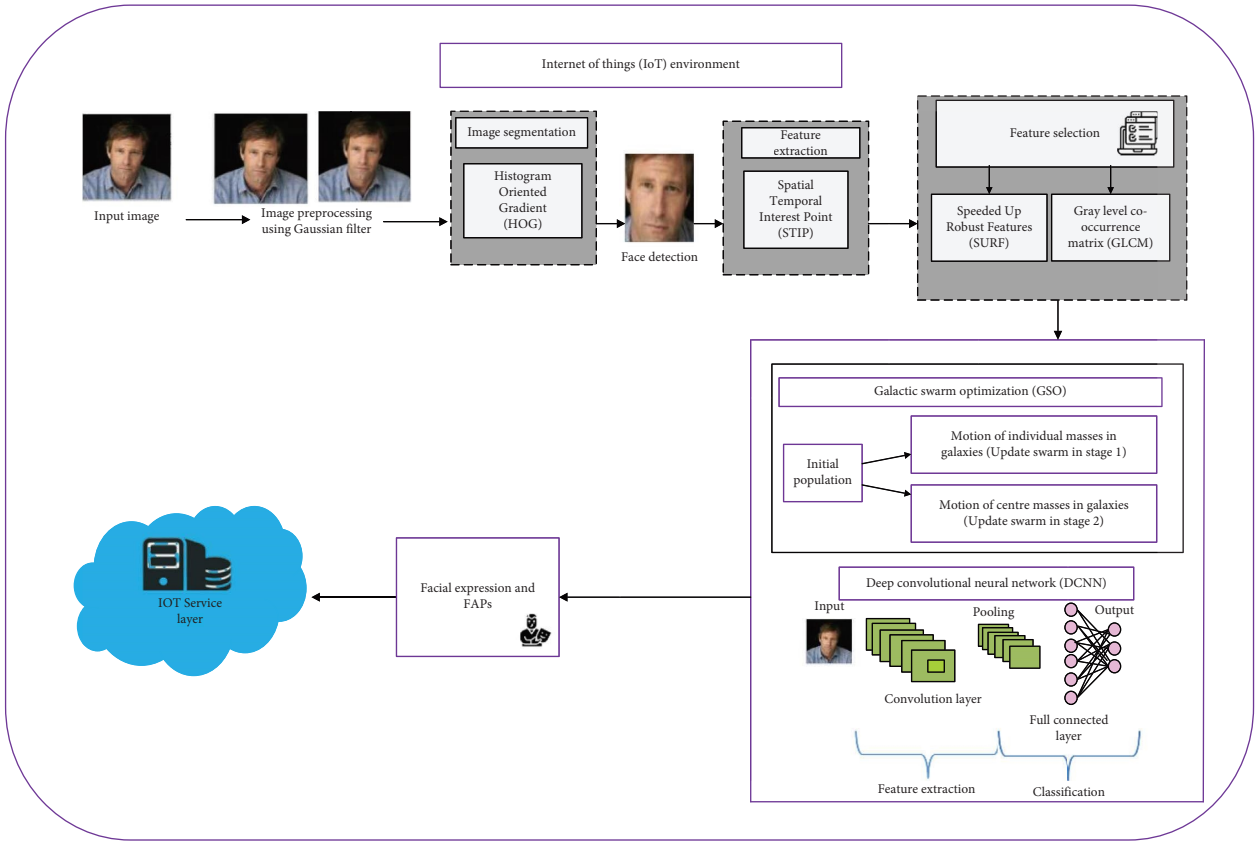

Figure 1: Proposed architecture (source: author).

that is convolved with the image. To produce the Gaussian kernel, the Gaussian function is evaluated at various positions in a matrix with the origin in the center. The convolution process on the image is then carried out using the obtained kernel. The Gaussian operators are convolution operators and the possibility of Gaussian smoothing is accomplished by convolution. It is a 2D convolution operator that is utilized for image smoothing and noise removal.

4.2. Image Segmentation Histogram of Oriented Gradients (HOGs). Region of interest segments the preprocessed image to find the appropriate region. HOG descriptors are widely used in image analysis and machine vision because of how well they work for face recognition. HOG can determine texture and shape by characterizing edge information via the gradient distribution of the local image. HOG-computing units include the cell and the block. First, we divide the picture into individual cells. Second, histograms are built from groups of cells. The shape descriptor is the final histogram averaged over all blocks. HOG feature extraction is shown in Figure 2. This can be done in three steps.

4.2.1. Step 1: Gradient Calculation. Calculating the horizontal and vertical gradients is necessary to obtain the gradient histogram. Assume that the pixel is in the coordinates (y, x) and that its grayscale value is denoted by the symbol $h(y, x)$. It may determine the gradients of the horizontal and vertical directions using the expressions $Hy(y, x)$ and $Hx(y, x)$, respectively [47].

$$Hy(y, x) = h(y + 1, x) - h(y - 1, x), \tag{2}$$

$$Hx(y, x) = h(y, x + 1) - h(y, x - 1). \tag{3}$$

Then, we may characterize the gradient strength and the gradient's directional bias as follows [47]:

$$N(y, x) = \sqrt{Hy(y, x)^2 + Hy(y, x)^2}, \tag{4}$$

$$C(y, x) = \arctan\left(\frac{Hx(y, x)}{Hy(y, x)}\right). \tag{5}$$

4.2.2. Step 2: Direction Vote. The orientations, whether signed or unsigned, are divided into L categories. Next, we use the pixel's gradient direction to establish its orientation bins. Finally, each bin's weight in the vote determines the pixel's gradient value. For the pixel with coordinates, we can write $N(y, x)$ for the value of the gradient and $C(y, x)$ for the direction of the gradient.

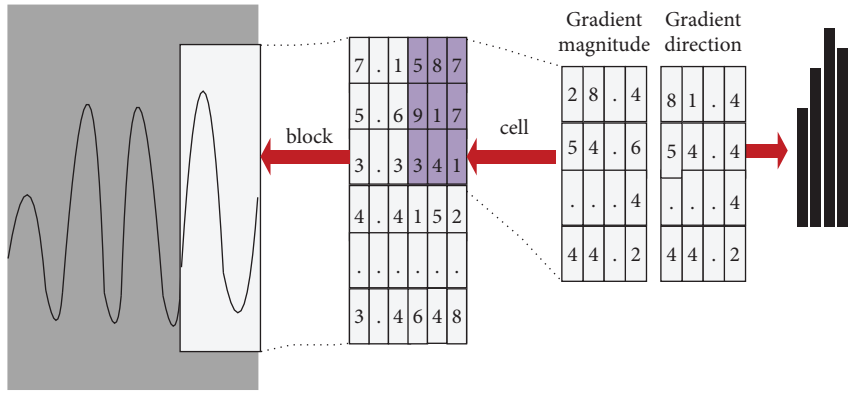

Figure 2: HOG feature extraction process [46].

$$e_{\text{bin}(j)} = N(y,x) \times \frac{C(y,x) - \text{bin}(j)}{K_{\text{bin}}}, \tag{6}$$

$$e_{\text{bin}(j+1)} = N(y,x) \times \frac{\text{bin}(j+1) - C(y,x)}{K_{\text{bin}}}, \tag{7}$$

where bin size in degrees of each direction is denoted by K_{bin} (for unsigned bins $= 180^{\circ}/K_{\text{bin}}$ and signed $K_{\text{bin}} = 360^{\circ}/K$), K_{bin} f represents the value-weighted assigned to orientation $K_{\text{bin}(j)}$ by the pixel (y,x), and $\text{bin}(j) \le C(y,x) < \text{bin}(j+1)$ are two nearby bins [47].

All of a cell's pixel gradient values are then voted into bins that correspond to the directions in which they are moving. This is how the histogram of gradients at various values becomes a descriptive statistic.

4.2.3. Step 3: Combination Histogram. Each cell's histogram is combined into a single vector representing the entire block. All of the blocks' related vectors are combined to form the image's feature descriptors.

$$\frac{b-a}{b+1} \times \frac{b-a}{a+1} \times a^2 \times L. \tag{8}$$

The full image feature descriptor has the length $(b-a)/(b+1) \times (b-a)$. The image is assumed to be segmented into $a \times a$ cells, each block to contain $b \times b$ cells, and no overlap to occur between blocks. HOGs also known as histogram of oriented gradients is the best object detection in image processing which uses the applications of feature descriptors. Fundamentally, it is the split of a single image into very small connected regions which are called cells, and for each cell, we compute a HOG direction. Each pixel of the cell provides gradient weights to its respective angular bin. We can take blocks as spatial regions, which are the neighboring cell groups. The basis for the classification and normalization of histograms is assembling cells as blocks. This process yields better invariance to changes in brightness or shadowing. Then, a two-dimensional facet model principle is also studied to detect the space-time interest point of an image. Finding the interest point in the video data depends on the change of gradient direction on a function under two dimensions is relatively large.

4.3. Feature Extraction Using Spatial-Temporal Interest Point. The fundamental idea behind STIP is to use the theory to determine the statistical qualities of an item and its low-level features, as well as to extract the temporal and spatial information of face recognition and its IoT environment. Each input image has uniformly sampled frames, and we use those frames to extract spatial-temporal information that will be used to represent an action. Particular spots (called interest points) are chosen for feature extraction because they experience the greatest fluctuation over time and space. The following sections detail the steps required to identify these hotspots and collect relevant features. Figure 3 depicts the spatial-temporal interest point of the image.

The linear scale representation $E(z,y,\sigma_u^2\tau_u^2)$ for an input image $U(z,w,s)$ is given by convolving a Gaussian kernel $H(z,w,\sigma_u^2\tau_u^2)$ with U as follows [48]:

$$E\left(z,y,\sigma_u^2\tau_u^2\right) = U(z,w,s) * H\left(z,w,\sigma_u^2\tau_u^2\right). \tag{9}$$

The spatial and temporal deviations are denoted by σ_u^2 and τ_u^2, respectively, while the " $*$ " represents a convolution operator. Finding points where U significantly varies in all three dimensions is the goal of Harris's 3D corner detector. The second-moment matrix s is convolved with a Gaussian function $h(z,w,\sigma_u^2\tau_u^2)$ of spatial and temporal variance to find these positions.

$$T = h\left(z,w,\sigma_u^2\tau_u^2\right) * \begin{bmatrix} E_Z^2 & E_zE_w & E_zE_s \\ E_zE_w & E_w^2 & E_wE_s \\ E_zE_s & E_wE_s & E_s^2 \end{bmatrix}. \tag{10}$$

To the extent that second-order derivatives can be expressed as follows [48]:

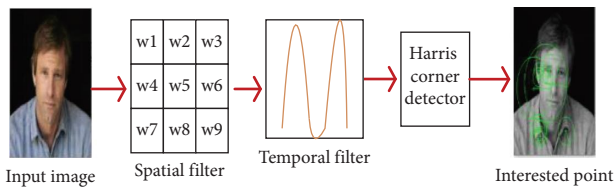

Figure 3: Spatial-temporal interest point of the image [48].

$$E_Z^2 = \frac{\partial^2 E}{\partial Z^2} \qquad E_w^2 = \frac{\partial^2 E}{\partial w^2} \qquad E_s^2 = \frac{\partial^2 E}{\partial s^2},$$
$$E_z E_w = \frac{\partial}{\partial z}\left(\frac{\partial E}{\partial W}\right) \quad E_w E_s = \frac{\partial}{\partial w}\left(\frac{\partial E}{\partial s}\right) \quad E_z E_s = \frac{\partial}{\partial z}\left(\frac{\partial E}{\partial s}\right). \tag{11}$$

The largest eigenvalues λ_1, λ_2, and λ_3 of an H represents a percentage of interest [48]. Harris corner function is used to identify this formula using

$$H = \lambda_1\lambda_2\lambda_3 - n(\lambda_1+\lambda_2+\lambda_3)^2. \tag{12}$$

Around these detected interest locations, visual and motion data are retrieved using local features like a histogram of oriented gradients (HOGs), together known as spatial-temporal interest points (STIPs) descriptors.

4.4. Feature Selection

4.4.1. SURF. Though the SURF approach shares some common ground with the SIFT framework, it adopts a slightly distinct set of principles. There is still a reliance on the old treat-the-scale space approach for detecting feature points. As a Hessian matrix on the scale, the coordinates (z, w) in the image are defined as follows [46]:

$$G = \begin{bmatrix} K_{zz}(\hat{z},\sigma)K_{zw}(\hat{z},\sigma) \\ K_{zw}(\hat{z},\sigma)K_{ww}(\hat{z},\sigma) \end{bmatrix}. \tag{13}$$

When $J = (z, w)$ is convolved with the Gaussian filtering function $h(\sigma) = 1/2\pi\sigma^{2^e} e^{-(z^2+w^2/2a^2)}$, the resulting function is the second derivative of the filter, whose significance is analogous to that of K_{zw} and K_{ww}.

The computational speed was increased by replacing the second-order Gaussian filter with a square filter approximation and speeding up the convolution using the integral image. In the original, expanding the box creates a Pyramid image with varying scales. Square filter template values C_{zz}, C_{zw}, and C_{ww} are muddled due to the image's complexity. In addition, it has determined the expression for the Gaussian matrix as follows [46]:

$$\triangle G = C_{zz}C_{ww} - (0.9C_{zw})^2. \tag{14}$$

Methods similar to SIFT are used to build image structures at different scales. Images with a four-step intensity scale are chosen at random. The size of the filter template may be seen in the grey area at the bottom. If the image is significantly larger than the reference, you can go ahead and bump up the purchase price. Hessian momenta correspond to scales if the filter template is $N * N$.

First, with the feature point as the origin, we rotate the coordinate axis to the main direction and then we choose the 20 s-long square region. The window space is broken up into 64 square-pixel portions. The wavelet response in the range of 5 s $*$ 5 s (s stands for the sampling step size) is computed for each subregion concerning the horizontal and vertical Haar in the principal direction. To make the response value more resistant to geometric modification, it is recorded as a weighted sum of two wavelet coefficients, c_z, and c_w. Each region's vectors are thus composed of four-dimensional elements [48].

$$Y_{sub} = \left(\sum c_z, \sum |c_z|, \sum c_w, \sum |c_w|\right). \tag{15}$$

This is achieved by first creating a normalized, illumination-invariant, 64-dimensional description vector for each feature point.

The steps involved in the SURF algorithm are as follows:

(a) Getting the input image.

(b) Finding the interesting point of detection: the blob detector detects the points of interest and stores them in the Hessian matrix.

(c) Describing the features: wavelet responses are used for describing the features.

(d) Matching the features: similar contrast feature, so fan object are compared.

(e) Detecting the objects: then, the relevant object is detected.

4.4.2. Gray Level Co-Occurrence Matrix (GLCM). The pixel of interest and its neighbours can be characterized by their distance and angle relationship using a technique called the gray level co-occurrence matrix (GLCM). Repeated distribution creates texture in the spatial position, so the distance and angle between two pixels are crucial.

The co-occurrence value is the distribution of co-occurrence values at a given distance (c) and angle (θ) from the pixel of interest, where $J(m,n)$ is the neighborhood of the pixel of interest. The co-occurrence matrix is defined for the $J(m,n)$ inequalities in the following equations [49]:

$$D_N = \sum_{m=1}^{l}\sum_{n=1}^{l}\begin{cases}1, & \text{if } J(m,n) = l \text{ and } J\left(m+c_z, n+c_w\right) = l,\\ 0, & \text{else,}\end{cases} \tag{16}$$

$$c_z = c.\cos(\theta), c_w = c.\sin(\theta). \tag{17}$$

Figure 4 shows the relationships between the images' distances and angles. Algorithm 1 depicts the GLCM.

The GLCM properties employed in this study, including contrast, homogeneity, correlation, and energy, are depicted in (9)–(12), one for each D_N and angle (θ) [49]. Table 2 expresses the angle of degree distances [50].

$$\underset{\text{contrast}}{\overset{\theta}{e}} = \sum_{j=1}^{K}\sum_{i=1}^{K}(j-i)^2 D_N\Big), \tag{18}$$

$$\underset{\text{homogenity}}{\overset{\theta}{e}} = \sum_{j=1}^{K}\sum_{i=1}^{K}\frac{D_N}{1+\left|j-i\right|}, \tag{19}$$

$$\underset{\text{correlation}}{\overset{\theta}{e}} = \sum_{j=1}^{K}\sum_{i=1}^{K}D_N\left[\frac{\left(j-\mu_j\right)\left(i-\mu_j\right)}{\sqrt{\left(\sigma_j\right)^2\left(\sigma_i\right)^2}}\right], \tag{20}$$

$$\underset{\text{energy}}{\overset{\theta}{e}} \sum_{j=1}^{K}\sum_{i=1}^{K}\left(D_N\right)^2\Big), \tag{21}$$

$$\mu_z = \sum_{j=1}^{K}\sum_{i=1}^{K}zD_N, \left(\sigma_z\right)^2 = \sum_{j=1}^{K}\sum_{i=1}^{K}D_N\left(z-\left(\mu_z\right)\right)^2, \tag{22}$$

where [50] σ_z is the standard deviation of GLCM and μ_z is the variance of GLCM.Once the boundary is marked, then the relevant features are selected using the gray level co-occurrence matrix (GLCM) which extracts the statistical texture features. It computes in rows and columns the number of gray levels G in an image. For a given neighbourhood, the pixel distance and intensity are calculated and thus help to change gray levels i and j. Due to their large dimensionality, the GLCMs are very sensitive to the size of the texture samples on which they are estimated. Thus, the number of gray levels is often reduced.

4.5. Face Recognition Classification

4.5.1. Galactic Swarm Optimization (GSO). Galactic swarm optimization is an optimization method that learns from animal herds, stars, galaxies, and other natural phenomena. In the second phase, the top performers from each subpopulation will form a new super swarm, and the GSO algorithm will return the leader of this swarm, who will represent the optimal solution discovered across the initial population (just as the PSO algorithm would have done after iterations).

GSO selects optimal weight parameters and dynamically modifies settings with DCNN. Gravity-driven stars and galaxies inspired this GSO algorithm. Exploration and exploitation are GSO phases: explore the vector space using subpopulation particles to find an ideal solution. The fittest subpopulation solution moves toward the global best through exploitation. Massive super cluster point masses are these galaxies. The super cluster is generated by massing these galaxies. GSO initially separates particles into M subswarms. PSO is done by subswarms. Equation (24) yields velocity and position updates [51].

$$U_i(l) \longleftarrow J_{V1}U(l) + D_1S_1\left(Q_{zi}(l) - f_i^{(l)}\right) \tag{23}$$

$$+ D_2S_2\left(H_z(l) - f_i^{(l)}\right), \tag{24}$$

$$f_i^{(l)} \longleftarrow f_i^{(l)} + U_i(l). \tag{25}$$

$U(l)$ Indicates the current velocity, $Q_{zi}(l)$ represents the personal best for particle $f_i^{(l)}$, and $H_z(l)$ represents the global subswarm solution. The direction of the fittest global and local solutions is represented by D_1 and D_2, inertia weight is J_{V1}, and S_1 and S_2 are given by the following formulas [52].

$$J_{V1} = 1 - \frac{n}{t_1 + 1}, \tag{26}$$

$$S_1 = \cup(-1,1), \tag{27}$$

where Iw1 is the inertia weight, m is the current iteration number 0–t_1, and S_1 and S_2 are random numbers. Global bests from subsequent stages are clustered to form superclusters. The following equation describes the superswarm W created by combining the global bests from Xk subswarms [51]:

$$w^{(l)}\epsilon W\colon l = 1,2,3,\ldots,M, \tag{28}$$

$$w^{(l)} = H_z(l). \tag{29}$$

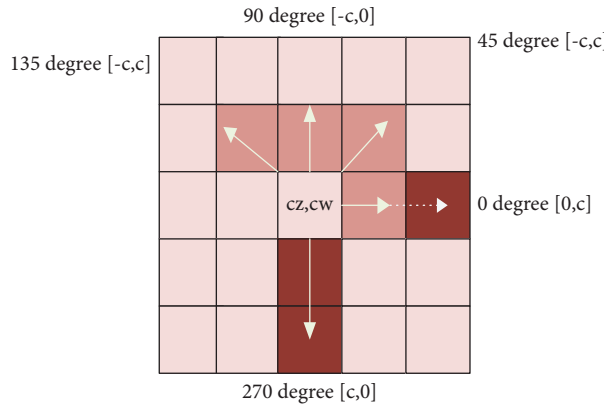

Figure 4: Distance and angle interactions between the image of interest and the environment [50].

Input: image.
Output: characteristics of the texture vector.
Begin
Step 1: obtain the GLACM matrix for the distance $c = 1$ along any of the four directions (0, 45, 90, and 135) by using the appropriate method.
Step 2: apply the standardization procedure for each GLCM matrix.
Step 3: each angle-specific GLCM matrix:
Step 3-1: texture characteristics can be calculated using their formulas.
Step 3-2: put the results of computations into a vector.
End

Algorithm 1: GLCM.

Table 2: Angular distances in degrees [50].

S. nos.	Expressed distance (c)	Angle degree (°)
1	0	0
2	−c, c	45
3	−c, 0	90
4	−c, c	135
5	c, 0	270

The second level, like the first, uses the PSO basis to compute particle position and velocity, but the equations have been somewhat modified in this level, as will be seen in the following examples [52].

$$U(l) \longleftarrow U_{w_2}(l) + D_3 S_3 \left(Q_j(l) - w^{(l)}\right), \tag{30}$$

$$+ D_4 S_4 \left(H_j - w^{(l)}\right), \tag{31}$$

$$w^l \longleftarrow w^l + U(l). \tag{32}$$

At this level, $Q_j(l)$ represents the personal best, D_3 and D_4 represent acceleration constants, and H_j represents the global best solution. The equations used in level 1 are used to estimate U_{w_2}, S_3, and S_4. It improves exploitation since super swarm focuses on the best global from subswarm. The superswarm exploits the computed information by using the best solution decided by the subswarms. To improve results, D_3 and D_4 parameters must be dynamically adjusted during execution. Face recognition adaptive parameters D_3 and D_4 are as follows:

$$D_3 = \frac{\sum_{i=1}^{S_{D3}} \mu_i^{D_3}(D_{3i})}{\sum_{i=1}^{S_{D3}} \mu_i^{D_3}},$$
$$D_4 = \frac{\sum_{i=1}^{S_{D4}} \mu_i^{D_4}(D_{4i})}{\sum_{i=1}^{S_{D4}} \mu_i^{D_4}}, \tag{33}$$

where S_{D3} and S_{D4} denote the whole rule of this face recognition system, D_{3i} and D_{4i} are the output result specifications for rule i, and $\mu_i^{D_3}$ *and* $\mu_i^{D_4}$ are the membership functions for rule i. The pseudocode for GSO is described in Algorithm 2.

Input
Initialization of Stage 1: $y_i^j, u_i^j q_i^j, h^j$ within the $[y_{\min}, y_{\max}]^S$ randomly.
Stage 2 is generated to evaluate stage 1: y^j, q^j, h^j uniformly inside the $[y_{\min}, y_{\max}]^S$.
For FQ $\leftarrow 1$ to $FQ_{\max}$
Begin PSO: Stage 1
f or $j \leftarrow 1$ *to* N **do**
f or $L \leftarrow 0$ *to* K_1 **do**
f or $j \leftarrow 1$ *to* M **do**
$u^j \leftarrow \omega_2 u^i + D_3 S_3 (q^j - x^i) D_4 S_4 (h - x^i) + D_2 S_2 (h^{(j)} - y_j^{(i)})$;
$y_j^{(i)} \leftarrow y_j^{(i)} + u_i^{(j)}$;
$q_i^{(j)} \leftarrow y_j^{(i)}$;
End if $e(y_j^{(i)}) < e(h^{(j)})$
then $h^{(j)} \leftarrow q_i^{(j)}$.
End if $e(h^{(j)}) < e(h)$
then $h \leftarrow h^j$.
End for
Begin PSO: Stage 2 flow with Stage 1
for $j \leftarrow 1$ *to* N
for $L \leftarrow 0$ *to* K_1 **do**
for $j \leftarrow 1$ *to* M **do**
$u^j \leftarrow \omega_2 u^i + D_3 S_3 (q^j - x^i) + D_4 S_4 (h - x^i)$
$x^i \leftarrow x^i + u^i$
End if $e(u^i) < e(q^i)$
then $q^i \leftarrow x^i$.
End if $e(q^i) < e(h)$,
then $h \leftarrow q^i$
End for
Output $h, e(h)$.

Algorithm 2: Pseudocode for GSO.

4.5.2. Deep Convolution Neural Network (DCNN). Computer images are made of pixels. They express visual data in binary. Its pixel value determines pixel brightness and hue. The human face processes a lot of information in the initial second of seeing an image.

(1) Convolution Layer. Face recognition uses different convolutional cores, and the deep CNN's convolutional layer receives higher layer output. The back propagation method fine-tunes the parameters of each convolutional unit in each convolutional neural network layer. The convolution operation separates input components. The initial convolution layer may miss simple features such as lines and corners. More network layers allow the recurrent extraction of complicated information from simpler ones. Figure 5 represents the convolutional layer.

The input image's fields of reception are checked repeatedly during the convolution procedure, resulting in several feature maps of the input data. The convolution layer is used for learning the convolution kernel's parameters, such as the weight matrix b and bias terms. Convolution's full computation formula is as follows [34]:

$$w_p = e \sum_{l=0}^{L-1} \sum_{k=0}^{K-1} z_{p+l+q+k} b_{lk} + c, \tag{34}$$

$$0 \le q \le P, \quad 0 \le q < Q. \tag{35}$$

Convolutional layers are used in deep neural networks, and their computation is based on linear correlations between input vectors of size (P and Q), where z the scalar value of interest and where activation is functions the value of interest.

(2) Pooling Layer. The network model's feature map can be reduced in size with the use of a nonlinear subsampling technique called the pooling layer. A pattern composed of many features can be extracted into a single feature and the next feature map can be constructed by convolving a set of existing feature maps. The feature map is then convolved and merged to produce a more complex map. In deep CNN, the pooling layer receives input from the preceding layer and subsamples all feature graphs, resulting in large decreases in input feature graphs. If the pooling layer uses uniform sampling and a 1×2 sampling size, the formula is as follows [34]:

$$w_{pq} = \frac{1}{t_1 t_2} \sum_{l=q}^{t_1-1} \sum_{k-1}^{t_2-1} z_{p^* t_{1+q^* t_{2+k}}}. \tag{36}$$

Following a pooling layer, the input and output values are denoted by z and w, respectively. A network segments the feature images of each layer into regions of a predetermined size and then determines the mean of the values in each section.

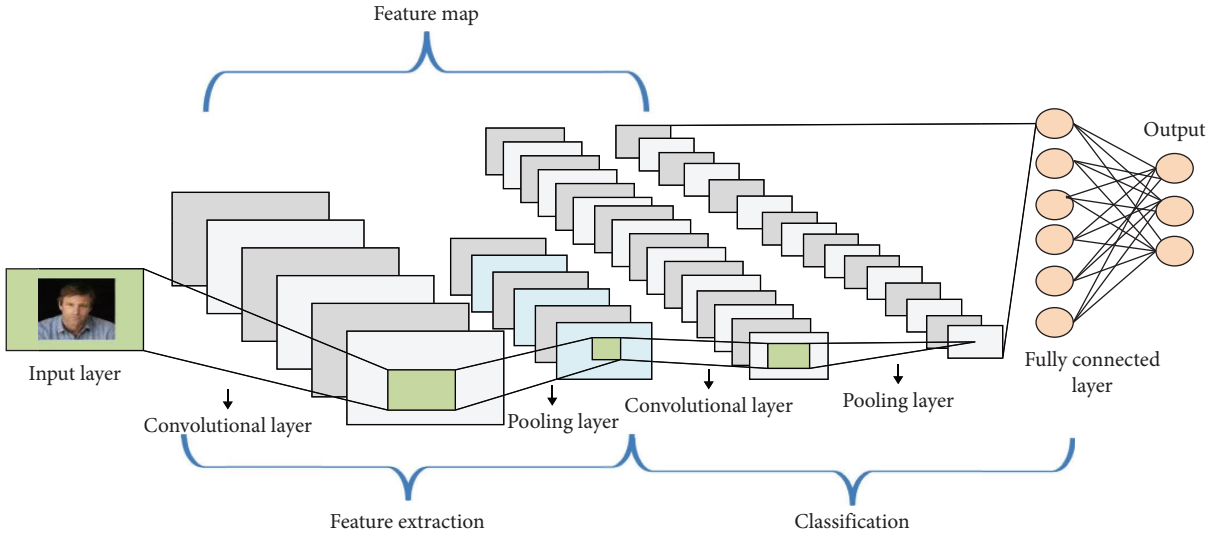

Figure 5: Convolutional layer [52].

The feature network in a DCNN is the mapping of features from the input image to the convolution kernel and activation function at the convolutional layer. The feature network group, created by superposing features from several convolutional kernels, represents various input image features [33].

$$z_l^i = e\left(\sum_{l=1}^{q_k^{l-1}} b_{l,k} \otimes z_k^{i-1} + c_k^i\right). \tag{37}$$

Add bias term c after convolution. The following is the equation used to determine neural output given a nonlinear activation function: the k^{th} feature graph of the current layer's additive bias is represented by c_k^i. While e is the activation function, c_k^i is normally initialized to 0 at the start.

The pooling layer is also known as the lowest sampling layer. By compressing data, the pooling layer maintains the number of feature graphs while reducing the storage space. Keeping the translation and scaling of the network model invariant is also possible. The two most commonly used pooling methods are maximum and average, which are basic in methodology. The max pooling procedure differs from the mean pooling operation in that the greatest pixel value must be recorded during the operation. The variable records the maximum value location for back propagation.

(3) Activation Unit. When neural networks use activation functions such as sigmoid and tanh, the resulting maps are nonlinear. A good mapping effect in the feature space is shown by a high value in the core region, which implies a large enhancement impact for the central signal but a relatively little influence on the outlying signals. The sigmoid and tanh functions can be represented by the following expressions [34]:

$$e(g) = sigmoid = \frac{1}{1 + \exp(-g)'}, \tag{38}$$

$$e(g) = \tanh(g) = \frac{f^g - f^{-h}}{f^g + f^{-g'}}. \tag{39}$$

In traditional neural network learning, the most important information is stored close to the network's center, while less relevant information is spread outwards. Each layer's output is proportional to the input received by the layer above it. In a neural network, no matter how many layers are present, the output is always just a linear combination of the input. The simplest perception extends the neuron by adding an activation function, making it nonlinear. Because the neural network can estimate nonlinear functions arbitrarily well, it can be used with a wide variety of nonlinear models. The following is the formula [34]:

$$e(z) = \max(0, z). \tag{40}$$

The outcomes of this analysis show that the ReLU function improves the network's recognition and learning efficiency. This is why most neural networks use the ReLU activation function.

(4) Fully Connected Layer. After the layered convolutional and pooling layers, DCNNs have one or more fully connected layers before the output layer. In a full-connection layer, all neurons communicate with their neighbours in the layer below them but not with one another. Neuron functions are comparable to human neuronal networks; they take input and output. In machine learning, a neuron is a mathematical function that simply takes an input and outputs. A classifier, full-connection layer enhances neural network nonlinear mapping and network size. Its mathematical expression is as follows [34]:

$$\varnothing_k^i = e\left(\sum_{l=1}^{q} z_l^{(i-1)} . b_{kl}^{(i)} + c_k^{(i)}\right). \tag{41}$$

where i is the number of layers that the moment.

The method used for determining the size of a hidden layer in a neural network is as follows [33]:

$$a_1^{(2)} = e\left(b_{11}^{(1)} z_1 + b_{12}^{(1)} z_2 + b_{13}^{1} z_3 + b_1^{(1)}\right), \tag{42}$$

$$a_2^{(2)} = e\left(b_{21}^{(1)} z_1 + b_{22}^{(1)} z_2 + b_{23}^{1} z_3 + b_2^{(1)}\right), \tag{43}$$

$$a_1^{(2)} = e\left(b_{31}^{(1)} z_1 + b_{32}^{(1)} z_2 + b_{33}^{1} z_3 + b_3^{(1)}\right), \tag{44}$$

$$g_{b,c}(z) = e\left(b_{11}^{(1)} a_1^{(2)} + b_{12}^{(1)} a_2^{(2)} + b_{13}^{(1)} a_3^{(2)} + b_1^{(1)}\right). \tag{45}$$

The formula is made more understandable by its streamlined expression. In particular, the formula for computing layer 2's unit can be written as follows, with symbols indicating the combined weights of layer 1's input layer 2's unit, and the offset items [34]:

$$g_l^2 = \sum_{k+1}^{q} b_{lk}^{(1)} z_k + c_l^{(1)}, a_l^{(2)} = e\left(x_l^{(2)}\right), \tag{46}$$

$$e\left([g_1, g_2, g_3]\right) = [e(g_1), e(g_2), e(g_3)]. \tag{47}$$

Parameters can be described as matrices, and the excitation function's expanded expression can be written as a vector (42). The forward propagation method of the neural network contains a set of equations that may be used to calculate the input and output of a neural network with a single hidden layer. Forward propagation algorithms rely heavily on finding the intermediate excitation value that corresponds to each layer. The following is a shorthand version of the expression [34]:

$$g^{(i+1)} = b^{(1)} a^{(1)} + c^{(1)}, \tag{48}$$

$$a^{(1+1)} = e\left(x^{i+1)}\right). \tag{49}$$

Suppose we have an excitation value for floor i, denoted by $g^{(i)}$, and an activation value for floor $i + 1$ denoted by $g^{(i+1)}$.

The number of layers in the neural network is denoted by $g^{(i)}$, the input layer is denoted by i, and the output layer is denoted by $g^{(i)}$. Figure 6 depicts the neural network forward propagation in which the network's primary concern is the determination of the excitation value at each hidden layer.

4.6. IoT Service Layer. Here, the extracted feature vectors and the facial action units (FAUs) are preserved in the cloud storage layer. The IoT service layer and the cloud storage layer are merged by the face image training process. The next section discusses implementation and experiments.

5. Experimental Result

The experimental examination of the proposed face recognition in an IoT environment for evaluation of performance is shown in this section. The findings indicate that the proposed method is highly effective. This subsection includes the dataset, simulation setup, comparison analysis, and research summary.

5.1. Dataset. The databases utilized in the studies are covered in this section. In the experiments and testing, databases were used. LFW are the databases. The large database known as Labelled Faces in the Wild (LFW) was created to identify faces in unrestricted settings. Two or more images of persons are present in those images. The images measure 250×250 pixels.

5.2. Simulation Setup. This subdivision provides examples of the simulation environment and setup for the suggested GSO-DCNN. Cloud storage preserves generated vectors of features and FAUs. The IoT service layer and cloud storage layer integrate for face image analysis. This proposed work's simulation output is applied in MATLAB R2020a. It has been demonstrated that our work obtains the best performance when the proposed framework is compared to several performance measures. Table 3 illustrates the simulation setup of the proposed work.

5.3. Comparative Analysis. In this research, the proposed method is compared to face plus support vector machine (face + SVM) [53], logistic regression-cat boost classifier-convolutional neural network (LR-CBC-CNN) [54], improved neural network (INN) [55], and convolutional neural network (CNN) in terms of accuracy, precision, recall, and error rate.

5.3.1. Accuracy. It evaluates the system's capacity to recognize various facial expressions and reach accurate findings. By dividing the number of faces that were properly identified by all the faces in the dataset, accuracy is commonly reported as a percentage. To identify the individual connected with a given face, the system must accurately match the face in question against a database of recognized faces. The proportion of successfully identified faces among all the faces evaluated serves as a gauge of identification accuracy. Accuracy is calculated using the following formula [36]:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}. \tag{50}$$

Figure 7 provides an illustration of the face recognition performance of the suggested, LR-CBC-CNN, and face + SVM algorithms. The accuracy in the proposed approaches is an average of 83.6%, face + SVM will achieve an

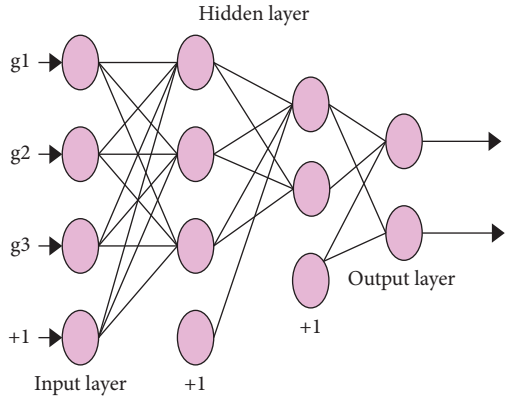

Figure 6: Structure of the hidden layer [35].

Table 3: System specifications (source: author).

Hardware configuration	Random access memory (RAM)	8 GB
	Hard disk	1 TB
	CPU processor	CPU: Intel(R) Core (TM) i5-4590S @ 3.00 GHz
Software Configuration	Running system	Windows Pro 10N
	Simulation tool	MATLAB R2020a

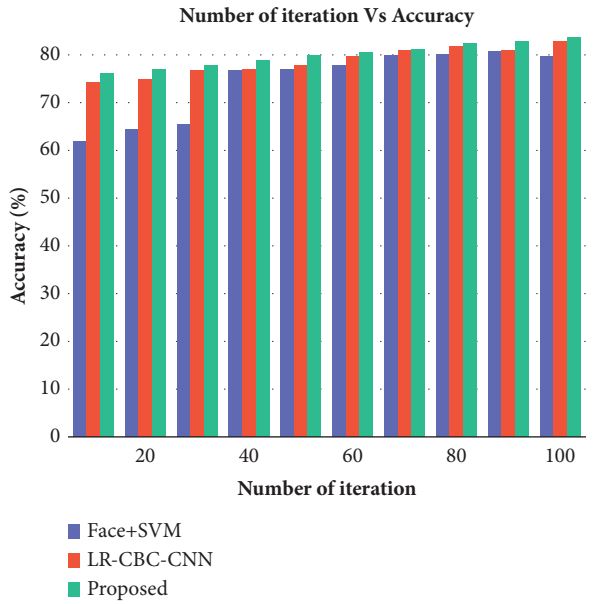

Figure 7: Number of iterations vs. accuracy (source: author).

average score of 82.8%, and LR-CBC-CNN will achieve a maximum accuracy of 79.7%. The suggested model for face detection has overall excellent outcomes.

5.3.2. Precision. In the field of face recognition, precision is a variable that assesses how well the system predicts positive outcomes. Precisely, accuracy is the ratio of true positives (cases correctly identified as positive) to total positives (true positives plus false positives). Precision is calculated using the following formula:

$$\text{Precision} = \frac{TP}{TP + FP}. \tag{51}$$

Figure 8 depicts the precision of the existing and proposed method. The proposed methods have an average precision of 72.2%, with face + SVM reaching 70.8% and LR-CBC-CNN peaking at 81.9%. When compared to standard methods such as face + SVM and LR-CBC-CNN, the proposed method presents better precision, and the model correctly identifies real-world faces.

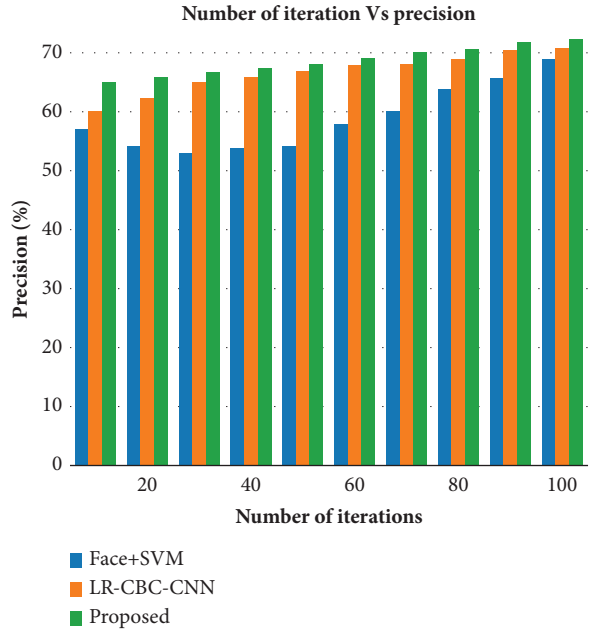

Figure 8: Number of iterations vs. precision (source: author).

5.3.3. Recall. Recall, frequently referred to as sensitivity or true positive rate, is a parameter used in face recognition that assesses how well the system can recognize every one of a positive category in comparison to the overall number of positive instances. The following formula is used to determine recall:

$$\text{Recall} = \frac{TP}{TP + TN}. \tag{52}$$

A high recall value means that a significant part of the positive examples is being successfully captured by the face recognition system, as shown in Figure 9. The strategies suggested have a maximum recall of 82.8% compared to existing methods including face + SVM getting an average of 80.2% and LR-CBC-CNN scoring a maximum of 81.9%. The proposed method has excellent recall when compared with the existing methods such as face + SVM and LR-CBC-CNN for recognizing faces.

5.3.4. Error Rate. The face expression "error rate" refers to a calculation of how much a model deviates from the true model in its predictions. When discussing classification models, the term "error rate" is frequently mentioned in models. Depending on the specific technology, algorithms, and datasets being used, the error rate for face recognition systems could vary significantly. The following formula is used to determine the error rate [36]:

$$\textit{Error rate} = \frac{1}{m}\sum_{k=1}^{m} Z_k - Z_k. \tag{53}$$

Figure 10 depicts the error rate of the existing and proposed method. Maximum scores for INN are 21.38 percent and for CNN, a total of 20.82 percent, while the error rate for the proposed methods is a maximum of 18 percent. The proposed experiment has a low error rate when compared to standard approaches such as INN and CNN.

6. Discussion

The suggested system was evaluated using evaluation criteria via experiments. Facial expressions provide valuable nonverbal information for studying human emotions and intentions. The technique given in this work improved human facial expression recognition performance and offered a novel solution to existing challenges in the literature. Our novel technique improves facial expression recognition and classification accuracy, recall, precision, and error rate, leading to better picture recognition. Our research indicates that GSO-CNN deep learning facial expression identification improves accuracy, interpretation, and efficiency, promoting prediction. Regarding the limitations of this research, the difficulty and time required for GSO-CNN model building but does not list particular problems. Discussing obstacles and solutions would improve research context comprehension.

6.1. Research Summary. The image dataset was collected from a public repository of a large database known as Labelled Faces in the Wild (LFW) and was created to identify faces in unrestricted environments. After that, preprocessing the image was performed by using a Gaussian filter to reduce noise and smooth it out into reality. Histogram of oriented gradients (HOG) is the gold standard for object detection in image processing using feature descriptor applications. The features associated with facial behaviors are extracted using spatial-temporal interest points (STIPs) in facial action units (FAUs). A powerful image-matching method called SURF is used for the efficient detection of the items. The detecting

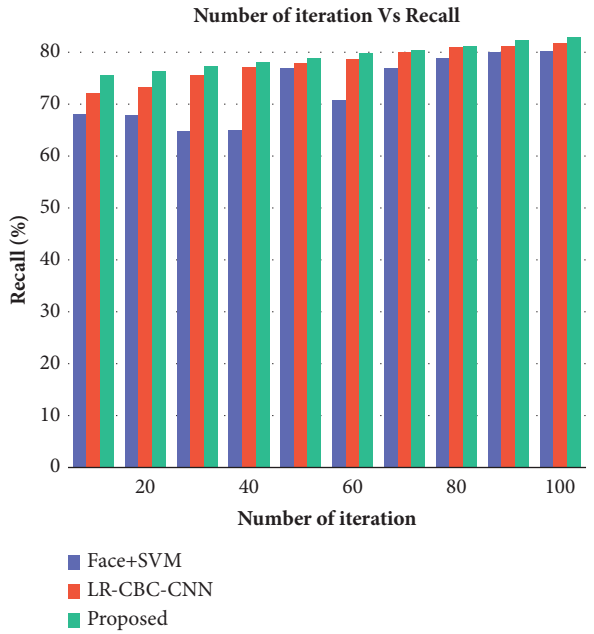

Figure 9: Number of iterations vs. recall (source: author).

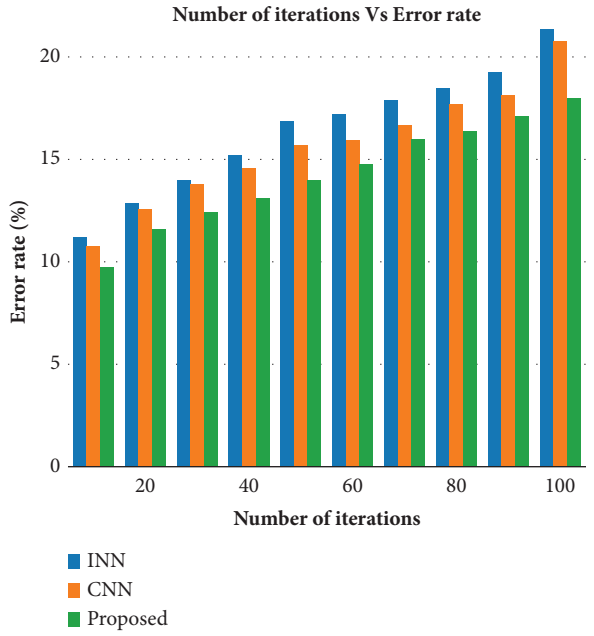

Figure 10: Number of iterations vs. error rate (source: author).

Table 4: Numerical outcomes (source: author).

Methods	Percentage of face recognition			
	Accuracy	Precision	Recall	Error rate
Face+ SVM	79.7	68.9	80.2	—
LR-CBC-CNN	82.8	70.8	81.9	—
INN	—	—	—	50.38
CNN	—	—	—	40.8
Proposed	83.6	72.2	82.8	18

performance was hindered by problems with feature selection and image matching. The gray level co-occurrence matrix (GLCM), which extracts random surface functions, was used to pick the relevant features. After that, we classified the images using galactic swarm optimization (GSO) algorithms based on CNN and identify any facial expressions. Finally, we evaluate the following performance metrics accuracy, precision, recall, and error rate to assess the suggested work. The discussion of the suggested approach's performance is included in this subsection. The findings of the comparison study are shown graphically in Figures 7–10, and Table 4 provides the numerical results of the comparative analysis.

7. Conclusion

In this research, we proposed the GSO-DCNN for face recognition in an IoT environment. The image dataset was collected from a publicly available source of LFW. The Gaussian filter technique is used for image preprocessing to remove noise and smooth the image, and the images are then processed in various ways for the HOG is used for image segmentation. STIP is used to extract features associated with face activities. Next, we used SURF to carry out the feature selection approach. The required features are extracted using a technique called GLCM, which classifies statistical texture features into distinct categories. The cloud service layer keeps the data preserved and safe. Metrics such as accuracy, precision, recall, and error rate were achieved by the proposed work. Deep learning model development, training, and deployment can be complex and time consuming. It might be difficult to create small efficient models while yet keeping high accuracy for use in Internet of Things (IoTs) applications due to technology limitations. In the future, we aim to improve performance and enlarge the scope of our experiments by investigating more deep learning techniques in a variety of applications.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] X. Su, Z. Wang, X. Liu, C. Choi, and D. Choi, "Study to improve security for IoT smart device controller: drawbacks and countermeasures," *Security and Communication Networks*, vol. 2018, pp. 1–14, 2018.

[2] S. Koppula and J. Muthukuru, "Secure digital signature scheme based on elliptic curves for internet of things," *International Journal of Electrical and Computer Engineering*, vol. 6, no. 3, p. 1002, 2016.

[3] J. Li, J. Wu, and L. Chen, "Block-secure: blockchain based scheme for secure P2P cloud storage," *Information Sciences*, vol. 465, pp. 219–231, 2018.

[4] G. Asharov, "Towards characterizing complete fairness in secure two-party computation," in *Proceedings of TCC*, pp. 291–316, Springer, 2014.

[5] X. Li and M. T. Orchard, "New edge-directed interpolation," *IEEE Transactions on Image Processing*, vol. 10, no. 10, pp. 1521–1527, 2001.

[6] X. Zhang and X. Wu, "Image interpolation by adaptive 2-d autoregressive modeling and soft-decision estimation," *IEEE Transactions on Image Processing*, vol. 17, no. 6, pp. 887–896, 2008.

[7] X. Liu, D. Zhao, S. M. RuiqinXiong, G. Wen, and H. Sun, "Image interpolation via regularized local linear regression," *IEEE Transactions on Image Processing*, vol. 20, no. 12, pp. 3455–3469, 2011.

[8] K. Guo, X. Yang, H. Zha, W. Lin, and S. Yu, "Multiscale semilocal interpolation with antialiasing," *IEEE Transactions on Image Processing*, vol. 21, no. 2, pp. 615–625, 2012.

[9] X. Liu, D. Zhao, J. Zhou, G. Wen, and H. Sun, "Image interpolation via graph-based bayesian label propagation," *IEEE Transactions on Image Processing*, vol. 23, no. 3, pp. 1084–1096, 2014.

[10] Z. A. Alizai, N. F. Tareen, and I. Jadoon, "Improved IoT device authentication scheme using device capability and digital signatures," in *2018 International Conference on Applied and Engineering Mathematics (ICAEM)*, pp. 1–5, IEEE, Beijing, China, 2018.

[11] X. Chen, H. Wang, Y. Liang, Y. Meng, and S. Wang, "A novel infrared and visible image fusion approach based on adversarial neural network," *Sensors*, vol. 22, no. 1, p. 304, 2021.

[12] Q. Tian, D. Han, K. C. Li, X. Liu, L. Duan, and A. Castiglione, "An intrusion detection approach based on improved deep belief network," *Applied Intelligence*, vol. 50, no. 10, pp. 3162–3178, 2020.

[13] S. Einy, C. Oz, and Y. D. Navaei, "IoT cloud-based framework for face spoofing detection with deep multicolor feature learning model," *Journal of Sensors*, vol. 2021, pp. 1–18, 2021.

[14] H. Hassani, C. Beneki, S. Unger, M. T. Mazinani, and M. R. Yeganegi, "Text mining in big data analytics," *Big Data and Cognitive Computing*, vol. 4, no. 1, p. 1, 2020.

[15] O. Haddad, F. Fkih, and M. N. Omri, "Toward a prediction approach based on deep learning in Big Data analytics," *Neural Computing & Applications*, vol. 35, no. 8, pp. 6043–6063, 2023.

[16] N. Kratzke, "A brief history of cloud application architectures," *Applied Sciences*, vol. 8, no. 8, p. 1368, 2018.

[17] C. Wang, S. Lee, and C. Lee, "Enabling declarative service composition for cloud applications," in *Proceedings of the 2016 IEEE International Conference on Network Infrastructure and Digital Content (IC-NIDC)*, pp. 186–189, New York, NY, USA, April 2016.

[18] D. Wu, L. Zhu, Q. Lu, and S. Sakr, "HDM: a composable framework for big data processing," *IEEE Transactions on Big Data*, vol. 4, no. 2, pp. 150–163, 2018.

[19] B. T. L. Liu and Y. Mao, "Declarative automated cloud resource orchestration," in *Proceedings of the 2nd ACM Symposium on Cloud Computing*, p. 26, ACM, Santa Cruz, CA, USA, November 2011.

[20] X. Zhang, C. Lee, and S. Helal, "Ipojo flow: a declarative service workflow architecture for ubiquitous cloud applications," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 4, pp. 1483–1494, 2019.

[21] N. Sahar, R. Mishra, and S. Kalam, "Deep learning approach-based network intrusion detection system for fog-assisted iot," in *Proceedings of the International Conference on Big Data,*

Machine Learning and Their Applications: ICBMA 2019, pp. 39–50, Springer, Singapore, November 2021.

[22] V. Pandimurugan, A. Jain, and Y. Sinha, "IoT based face recognition for smart applications using machine learning," in *Proceedings of the 2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)*, pp. 1263–1266, IEEE, Thoothukudi, India, December 2020.

[23] X. Yi, H. Pan, H. Zhao et al., "Cycle generative adversarial network based on gradient normalization for infrared image generation," *Applied Sciences*, vol. 13, no. 1, p. 635, 2023.

[24] T. Hou, H. Xing, X. Liang, X. Su, and Z. Wang, "A marine hydrographic station networks intrusion detection method based on LCVAE and CNN-BiLSTM," *Journal of Marine Science and Engineering*, vol. 11, no. 1, p. 221, 2023.

[25] T. A. Kumar, R. Rajmohan, M. Pavithra, S. A. Ajagbe, R. Hodhod, and T. Gaber, "Automatic face mask detection system in public transportation in smart cities using IoT and deep learning," *Electronics*, vol. 11, no. 6, p. 904, 2022.

[26] P. M. Kumar, U. Gandhi, R. Varatharajan, G. Manogaran, T. Vadivel, and T. Vadivel, "Retracted article: intelligent face recognition and navigation system using neural learning for smart security in Internet of Things," *Cluster Computing*, vol. 22, no. S4, pp. 7733–7744, 2019.

[27] A. Rahim, Y. Zhong, T. Ahmad, S. Ahmad, P. Pławiak, and M. Hammad, "Enhancing smart home security: anomaly detection and face recognition in smart home IoT devices using logit-boosted CNN models," *Sensors*, vol. 23, no. 15, p. 6979, 2023.

[28] M. Masud, G. Muhammad, H. Alhumyani et al., "Deep learning-based intelligent face recognition in IoT-cloud environment," *Computer Communications*, vol. 152, pp. 215–222, 2020.

[29] E. J. Cheng, K. P. Chou, S. Rajora et al., "Deep sparse representation classifier for facial recognition and detection system," *Pattern Recognition Letters*, vol. 125, pp. 71–77, 2019.

[30] G. Rajeshkumar, M. Braveen, R. Venkatesh et al., "Smart office automation via faster R-CNN based face recognition and internet of things," *Measurement: Sensors*, vol. 27, 2023.

[31] M. Umer, S. Sadiq, R. M. Alhebshi et al., "Face mask detection using deep convolutional neural network and multi-stage image processing," *Image and Vision Computing*, vol. 133, 2023.

[32] M. Abdel-Basset, H. Hawash, V. Chang, R. K. Chakrabortty, and M. Ryan, "Deep learning for heterogeneous human activity recognition in complex iot applications," *IEEE Internet of Things Journal*, vol. 9, no. 8, pp. 5653–5665, 2022.

[33] V. Muthiah-Nakarajan and M. M. Noel, "Galactic swarm optimization: a new global optimization metaheuristic inspired by galactic motion," *Applied Soft Computing*, vol. 38, pp. 771–787, 2016.

[34] W. Li, J. Li, and J. Zhou, "Deblurring method of face recognition AI technology based on deep learning," *Advances in Multimedia*, vol. 2022, Article ID 9146711, 9 pages, 2022.

[35] M. Mandal, "Introduction to convolutional neural networks (CNN)," 2024, https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/.

[36] K. Okokpujie, S. John, C. Ndujiuba, J. A. Badejo, and E. N. Osaghae, "An improved age invariant face recognition using data augmentation," *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 1, pp. 179–191, 2021.

[37] G. Meena, K. K. Mohbey, A. Indian, M. Z. Khan, and S. Kumar, "Identifying emotions from facial expressions using a deep convolutional neural network-based approach," *Multimedia Tools and Applications*, vol. 83, no. 6, pp. 15711–15732, 2023.

[38] K. Albulayhi and Q. A. Al-Haija, "Adversarial Deep Learning in Anomaly Based Intrusion Detection Systems for IoT Environments," *International Journal of Wireless and Microwave Technologies*, vol. 13, 2023.

[39] J. Shao and Y. Qian, "Three convolutional neural network models for facial expression recognition in the wild," *Neurocomputing*, vol. 355, pp. 82–92, 2019.

[40] M. Yang, X. Wang, G. Zeng, and L. Shen, "Joint and collaborative representation with local adaptive convolution feature for face recognition with single sample per person," *Pattern Recognition*, vol. 66, pp. 117–128, 2017.

[41] J. Yu, K. Sun, F. Gao, and S. Zhu, "Face biometric quality assessment via light CNN," *Pattern Recognition Letters*, vol. 107, pp. 25–32, 2018.

[42] H. Hu, S. A. A. Shah, M. Bennamoun, and M. Molton, "2D and 3D face recognition using convolutional neural network," in *Proceedings of the TENCON 2017-2017 IEEE Region 10 Conference*, pp. 133–132, Penang, Malaysia, December 2017.

[43] Z. Lu, X. Jiang, and A. Kot, "Deep coupled resnet for low-resolution face recognition," *IEEE Signal Processing Letters*, vol. 25, no. 4, pp. 526–530, 2018.

[44] Y. Zhong, S. Oh, H. C. Moon, Y. FuturZhong, S. Oh, and H. C. Moon, "Service transformation under industry 4.0: investigating acceptance of facial recognition payment through an extended technology acceptance model," *Technology in Society*, vol. 64, Article ID 101515, 2021.

[45] G. F. Plichoski, C. Chidambaram, and R. S. Parpinelli, "A face recognition framework based on a pool of techniques and differential evolution," *Information Sciences*, vol. 543, pp. 219–241, 2021.

[46] M. Tyagi, "HOG (histogram of oriented gradients): an overview," *Data Science*, 2021, https://towardsdatascience.com/hog-histogram-of-oriented-gradients-an-overview-7a4a6225e34.

[47] K. Chen, S. Chen, S. Zhang, and H. Zhao, "Automatic modulation recognition of radar signals based on histogram of oriented gradient via improved principal component analysis," *Signal, Image and Video Processing*, vol. 17, no. 6, pp. 3053–3061, 2023.

[48] T. Zhang, R. Zhao, and Z. Chen, "Application of migration image registration algorithm based on improved SURF in remote sensing image mosaic," *IEEE Access*, vol. 8, Article ID 163637, 2020.

[49] R. Lionnie, C. Apriono, and D. Gunawan, "Face mask recognition with realistic fabric face mask data set: a combination using surface curvature and glcm," in *Proceedings of the 2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*, pp. 1–6, Toronto, Canada, April 2021.

[50] S. Yenduri, V. Chalavadi, and C. K. Mohan, "STIP-GCN: space-time interest points graph convolutional network for action recognition," in *Proceedings of the 2022 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, Padua, Italy, July 2022.

[51] S. Saha, "A comprehensive guide to convolutional neural networks— the ELI5 way Data Science," 2018, https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53.

[52] R. Silwal, A. Alsadoon, P. W. C. Prasad, O. H. Alsadoon, and A. Al-Qaraghuli, "A novel deep learning system for facial feature extraction by fusing CNN and MB-LBP and using enhanced loss function," *Multimedia Tools and Applications*, vol. 79, no. 41-42, pp. 31027–31047, 2020.

[53] Y. Zheng, H. Wang, and Y. Hao, "Mobile application for monitoring body temperature from facial images using convolutional neural network and support vector machine," *Mobile Multimedia/Image Processing, Security, and Applications 2020*, vol. 11399, pp. 53–63, 2020.

[54] S. Zhang, X. Lu, and Z. Lu, "Improved CNN-based CatBoost model for license plate remote sensing image classification," *Signal Processing*, vol. 213, 2023.

[55] Z. Chen, A. Huang, and X. Qiang, "Improved neural networks based on genetic algorithm for pulse recognition," *Computational Biology and Chemistry*, vol. 88, 2020.