Computer Communications 152 (2020) 215–222

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Deep learning-based intelligent face recognition in IoT-cloud environment

Mehedi Masud [a], Ghulam Muhammad [b,*], Hesham Alhumyani [a], Sultan S Alshamrani [a], Omar Cheikhrouhou [a], Saleh Ibrahim [c,d], M. Shamim Hossain [e]

[a] College of Computers and Information Technology, Taif University, Taif, Saudi Arabia
[b] Department of Computer Engineering, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia
[c] Electrical Engineering Department, Taif University, Saudi Arabia
[d] Computer Engineering Department, Cairo University, Egypt
[e] Department of Software Engineering, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia

ARTICLE INFO

Keywords:
Deep neural network
Intelligent face recognition
Healthcare-IoT
Cloud environment

ABSTRACT

In recent years, the Internet-of-Things (IoT) technology is being used in many application areas such as healthcare, video surveillance, transportation etc. The massive adoption and growth of IoT in these areas are generating a massive amount of data. For example, IoT devices such as cameras are generating a huge amount of images when used in hospital surveillance scenarios. Here, face recognition is an important element that can be used for securing hospital facilities, emotion detection and sentiment analysis of patients, detecting patient fraud, and hospital traffic pattern analysis. Automatic and intelligent face recognition systems have high accuracy in a controlled environment; however, they have low accuracy in an uncontrolled environment. Also, the systems need to operate in real-time in many applications such as smart healthcare. This paper suggests a tree-based deep model for automatic face recognition in a cloud environment. The proposed deep model is computationally less expensive without compromising the accuracy. In the model, an input volume is split into several volumes, where a tree is constructed for each volume. A tree is defined by its branching factor and height. Each branch is represented by a residual function, which is constituted by a convolutional layer, a batch normalization, and a non-linear function. The proposed model is evaluated in various publicly available databases. A comparison of performance is also done with state-of-the-art deep models for face recognition. The results of the experiments demonstrate that the proposed model achieved accuracies of 98.65%, 99.19%, 95.84% on FEI, ORL, and LFW databases, respectively.

1. Introduction

The introduction of many Internet of Things (IoT) and smart body sensors has increased the volume of data significantly in recent years. The nature of the data is heterogeneous and sparse in most of the cases. The processing of Big Data is a matter of concern for real-time applications [1–3]. Consider a scenario where a person is to be recognized in an airport where there are many sensor cameras. These cameras capture images of many objects including humans in their focus areas, and these are capturing images continuously. This huge amount of image data should be processed in a meaningful way in a cloud environment so that a specified person can easily be recognized.

Face recognition is one of the oldest yet a dynamic topics of research. It is necessary for security and biometric applications. Early face recognition systems relied on manual features and traditional classifiers. Some hand-crafted features include local binary pattern (LBP), Weber local descriptors (WLD), principal component analysis (PCA), and histogram of oriented gradients (HOG). Traditional classifiers include support vector machines (SVM), linear discriminant analysis (LDA), and some minimum distant-based classifiers. These features and classifiers work well in a controlled environment, where faces are mostly frontal and with a neutral expression, and having less variation of illumination. However, in many applications such as those related to surveillance, face images may be occluded, not frontal, and having low resolution and high variation of illumination. For these applications, the traditional face recognition systems may not work properly.

Deep learning is a powerful machine learning technique that has been successfully used in many signal processing applications. The applications include speech and speaker recognition, image recognition [4], and video recognition. Since the introduction of deep learning, many architectures have been proposed in the literature. These architectures differ in many aspects such as the number of layers, the number of filters, the size of filters, and the arrangement of layers. If the number of layers is high, the architecture can be called deep, otherwise, it can be called shallow. If the filter size is big, the architecture is called wide, otherwise, it is thin. Normally, a deep architecture provides a

[*] Corresponding author.
E-mail address: ghulam@ksu.edu.sa (G. Muhammad).

https://doi.org/10.1016/j.comcom.2020.01.050
Received 30 September 2019; Received in revised form 8 January 2020; Accepted 23 January 2020
Available online 27 January 2020

high accuracy, which can be evident in popular GoogleNet [5], VGG Net [6], AlexNet [7], and ResNet [8]. These architectures are also thin. On the other hand, shallow and wide architecture such as a wide residual network (WRN) emphasized filter size rather than the depth of the network [9]. Each of these architectures has its own advantages and disadvantages. A very deep and thin network may achieve good performance; however, it will have a very high number of parameters. Normally, a deep learning network with a high number of parameters may not fit for a real-time application. Therefore, there needs an equilibrium between the precision and the parameters needed for security-related real-time applications [10].

The applications in a smart city need an accurate output in real-time. For example, the traffic congestion and the alternative routes should be determined in real-time, secured access needs accurate verification, and a medical diagnosis should be error-free [11,12]. The smart city may have many components such as smart homes [13], smart healthcare [14,15], smart schools, smart connected vehicle [16], and smart shopping [17]. Due to an increase of IoTs, data volume has been increased by manifolds. This volume is considered as heterogeneous, and unfiltered. This huge amount of data should be processed carefully so that the output is accurate, security is not breached, and the processing does not take much time. It is very difficult to achieve all these things at a time; however, there should be a balance between them [18].

The face is an important biometric component in human verification and recognition. It can be used as a secured verification process. Many face recognition systems have been proposed for the last few decades. The face recognition research has evolved from recognizing faces taken in a controlled environment to faces captures in an uncontrolled environment, from traditional feature extraction techniques to deep-learned feature extraction techniques.

In this paper, we review some face recognition systems based on deep neural networks. Then, we propose a new face recognition system using a tree-based deep neural network [19–21]. The tree-based neural network provides a good accuracy using a respectable number of parameters; therefore, it works in real-time, which is very important for any secured smart city applications. This is the first time the tree-based deep network is used in the face recognition system.

The major contributions of the paper are: (i) presenting the tree-based deep model for automatic face recognition system; (ii) evaluating the system on three public face databases; and (iii) comparing the system with state-of-the-art face recognition systems, and achieving better performance, and (iv) proposing a smart city framework where the system can be deployed.

The paper is structured in the following manner. Section 2 discusses some related works in the literature. Section 3 presents the smart city environment, the tree-based deep model for face recognition, and a description of the databases. Section 4 provides experimental setup, results and discussion. Finally, Section 5 draws some conclusions.

2. Related studies

In this section, the advancement of face recognition using deep learning is discussed. In almost all the related previous works used convolutional neural networks (CNN) as the main architecture. A light CNN with max-feature-map units was used in biometric face recognition applications in [22]. A biometric quality assessment method was embedded in the CNN. Sun et al. [23] used cascaded restricted Boltzmann machines to form a deep convolutional network for face verification. The method achieved a moderate accuracy of 93.83% using labeled faces in the wild (LFW) database.

Deep CNN-based face recognition for infants was proposed in [24]. In the database, there were 2100 face images of 210 infants, where each infant had 10 images. The authors found that increasing the number of layers does not perform well in infant face recognition. In the experiments, they found 91.03% accuracy. Guo et al. [25] designed a deep network using the VGGNet by fusing features from the visible light image and near-infrared image for face recognition. A fusion strategy was proposed to fuse scores. They achieved 97.35% accuracy using the LFW database.

Table 1
Summary of previous related works.

Ref.	Model name	Database	Acc. (%)
[22]	Light CNN	CASIA, FLW	99.0
[23]	CNN-RBM	LFW	93.8
[24]	Deep CNN	Private, Newborn	91.0
[25]	VGGNet	LFW, YTF	97.3
[26]	CNN-2	ORL	95
[27]	CNN	RGB,D,T	–
[28]	Deep coupled ResNet	LFW	99
[29]	Deep face	LFW, YTF	97.3
[30]	Deep ID	LFW	97.4
[31]	Deep ID2	LFW	99.5
[32]	VGGFace	LFW	98.9
[33]	FaceNet	LFW	99.6
[34]	AMS loss, Caffe	LFW	94.5
[35]	CosFace	LFW	99.3

Hu et al. [26] studied the performance of 2D and 3D face recognition using two different models of CNN. They found that the deeper CNN model performed better than the other CNN model. In the experiments, 95% accuracy was obtained using the Olivetti Research Laboratory (ORL) database. A multimodal face recognition system using modality-specific (RGB and depth) CNNs was proposed in [27]. Some hand-crafted features such as LBP, HOG, and Haar-like features were also fused to the deep-learned features to improve the performance.

A deep coupled ResNet (DCR) model consisting of one trunk network and two branch networks were proposed for face recognition in [28]. The branch networks were used to convert high-resolution images into intended low-resolution images. The model achieved 99% accuracy in the LFW database.

Some popular models for face recognition include DeepFace [29], DeepID, DeepID2, and DeepID2+ models. Taigman et al. developed the DeepFace model, which has eight layers. The first three layers have convolution-pooling layers, the next three layers are locally connected layers, while the last two layers are fully connected layers. On the LFW database, the model got more than 97% accuracy. DeepID and its variants used an ensemble of small CNNs and fused them. Each small CNN had four convolutional layers, three pooling layers and two fully connected layers [30,31]. The variations occurred in the number of filters in the layers. 99.47% accuracy was obtained by the DeepID2+ model using the LFW database.

The VGGFace introduced in 2015 used the VGGNet-16 model, and obtained around 99% accuracy on the LFW database [32]. Google introduced the FaceNet [33], which used GoogleNet-24 architecture, got more than 99% accuracy on the same database. ResNet-based face recognition systems such as AMS loss [34] and CosFace [35] were recently developed. These systems achieved high accuracy in several databases. Table 1 gives a summary of the previous works.

Almost all of the models mentioned in the section mainly focused on improving the accuracy of face recognition in different constraints. Very few attempted to develop a low-complex deep model for face recognition.

3. Materials and methods

3.1. Smart city framework

The smart city framework consists of several components including smart homes, smart traffic systems, smart shopping, smart healthcare, high-speed wireless networks, and cloud servers. Fig. 1 shows the smart city framework used in this work. Various signals captured by IoTs

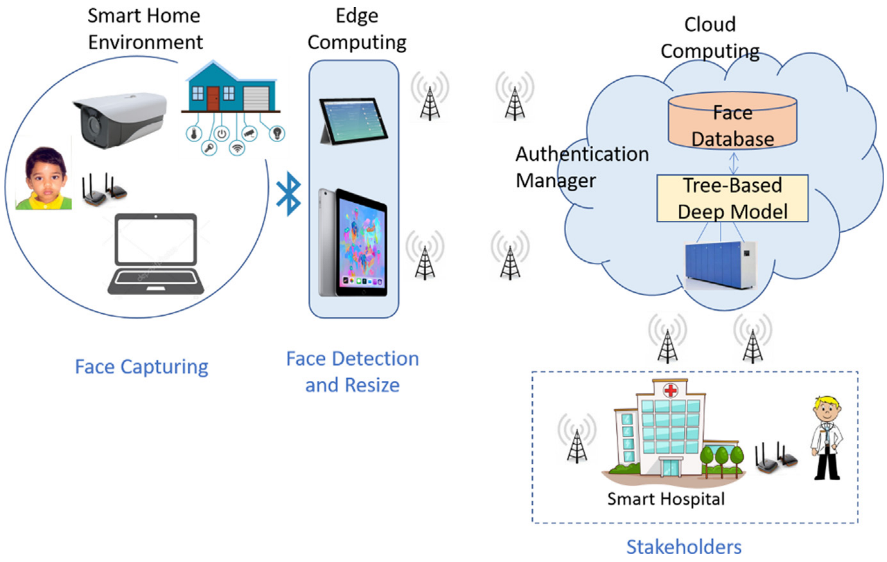

Fig. 1. Structure of a smart city framework for face recognition.

and smart devices are sent to the cloud server for processing and decision. After processing, the decision is sent to the stakeholders. Stakeholders then can take necessary actions accordingly. Suppose a scenario where a person wants to access a park. A smart device captures the person's facial image and sends the image to the cloud using 5G wireless technology.

In the cloud, there are several virtual machines (VMs) that can work in parallel. These VMs are equipped with high processing power. The face image is processed in real-time in one of these VMs, and verification is made. The decision is sent to the gate of the park whether the person can access or not to the park. The cloud has large memory storage that can store a bulk of multimedia data. The VMs can run in parallel so that the cloud can handle multiple requests at the same time [36].

The framework may consist of an edge computing facility. The edge computing is placed at the edge of local machines and before the radio access networks [37,38]. The purpose of the edge computing is to process and filter signals so that the volume of data that should be transmitted to the cloud is minimized. The local machines in the edge computing can be smartphones or devices that communicate between themselves to minimally process the signals. These devices have the low processing power and their life is restricted by the operating batteries. The task of processing is distributed to the devices using an optimization algorithm [39]. The attributes that are taken into account for the optimization are the battery life, the processing power, the current load, and the volume of the data. Once the processing is finished, the processed data are sent to the cloud.

The cloud may have a cloud manager, whose main tasks include authenticating a request and distributing the processing to the VMs. For the classification purpose, the manager consults with the storage of the cloud for pertained parameters. Once the classification is finished, the result is sent to the stakeholders via the cloud manager.

3.2. Tree-based deep network

A deep neural network (DNN) has fetched many advantages in machine learning. Since its invention, the DNN has been used in numerous applications especially in image and speech signal processing. The DNN has achieved very high accuracy in applications related to image and speech processing. The deep network requires a large amount of data for training.

There are many architectures of the DNN. Some of them focus on the depth while others focus on the size of the filters. Depending on the application, we may use one of these types of architecture. Recently, the tree-based deep network has been proposed [19]. The idea of the tree-based deep model is to distribute the processing in a tree-like structure.

Two deep models are proposed in this paper. These models are a single tree model and a parallel tree model. First, the single tree model structure is described.

In the single tree model, first, the input volume is mapped to a new volume by increasing the number of channels. Suppose that the new volume is H × W × C, where H × W is the spatial dimension of the image, and C is the channel number. This volume is split to a number of groups equaling the branching factor (b) of the tree. A residual function is applied to each member of the groups. The residual function, as shown in Fig. 2(a), is the basic building block of the tree-based deep model [40]. The residual function has three operations in succession, which are convolution, batch normalization, and a non-linear activation function in the form of a rectifier linear unit (ReLU). This residual function is applied to each member of the groups. At each tree node, the same splitting algorithm is applied to construct a tree until a prescribed number of tree height (l) is reached.

In the case of the parallel tree model, once the input volume is mapped to H × W × C, it is split into g number branches. The volume of each of the split ones is H × W × C/g. Now g number of parallel trees is constructed the same way as the single tree model. Each of the parallel trees is a single tree. The volume H × W × C/g is passed through a residual function, followed by a single tree structure. Therefore, the parallel tree model is an extension of the single tree model. Fig. 3 shows the architecture of the parallel tree model.

The tree-based deep model is beneficial to a smart city environment. It has several advantages over the traditional deep models. The advantages include (i) a balance between the number of parameters and accuracy, (ii) less computational time compared with other very deep models, (iii) parallel computation of trees, and (iv) a high information density.

The outputs of branches of a tree are concatenated to produce the volume of the original input to the tree. Another residual function is then applied to the concatenated output. For the parallel trees, if we have four parallel trees, we have four concatenated outputs followed by corresponding residual functions. The outputs are then again concatenated to yield the final volume.

Fig. 2(b) displays the structure of deep models. Both the models (single tree and parallel trees) are designed into three stages. A convolution filter is realized to the input volume of the image to convert into an

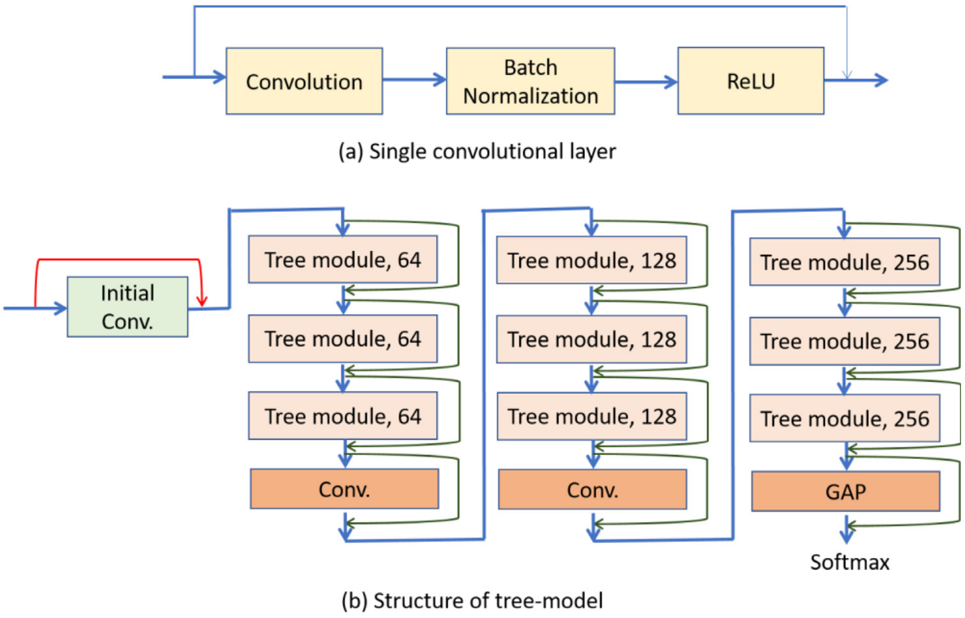

Fig. 2. (a) The components of the residual function, and (b) the structure of the tree-based deep model.

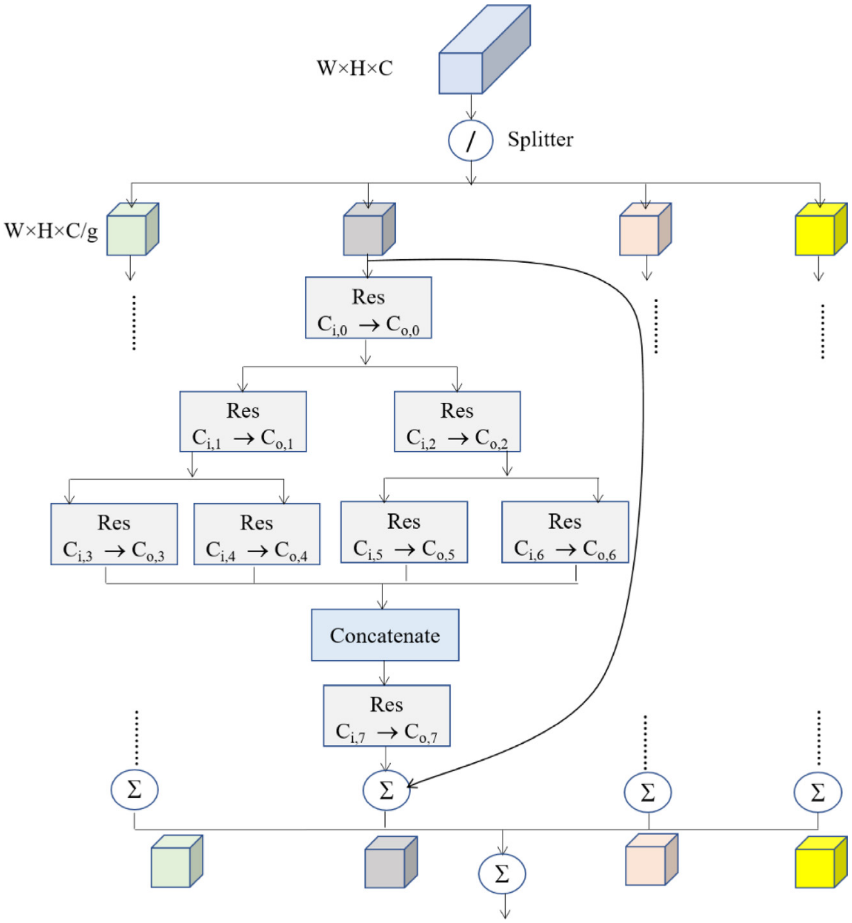
W×H×C
Splitter
W×H×C/g
Res $C_{i,0} \rightarrow C_{o,0}$
Res $C_{i,1} \rightarrow C_{o,1}$
Res $C_{i,2} \rightarrow C_{o,2}$
Res $C_{i,3} \rightarrow C_{o,3}$
Res $C_{i,4} \rightarrow C_{o,4}$
Res $C_{i,5} \rightarrow C_{o,5}$
Res $C_{i,6} \rightarrow C_{o,6}$
Concatenate
Res $C_{i,7} \rightarrow C_{o,7}$
Σ

Fig. 3. The architecture of the parallel tree model.

image of a predetermined number of channels. This new image is the into to the first stage. Each stage has three tree modules as shown in the figure. The number of filters in a stage is doubled by the number of filters in the previous stage. The first stage has 64 convolutional filters. Because of the independent nature of the tree modules, the convolution operations in these modules can run in parallel to reduce the overall execution time.

The stages bring in flexibility in the model in the sense that one stage structure does not depend on other stages when hyperparameters vary. An increase in the number of channels will increase the number of parameters, which may cause overfitting. Therefore, we need to downsample. Between the stages, there are convolutional layers whose job is to reduce the number of parameters. This is done by choosing a stride equal to two, which is equivalent to a downsample by a factor of two. This type of downsampling is superior to a pooling layer. The output maps of these three stages are 32, 16, and 8, respectively. The number of channels of these stages is 64, 128, and 256, respectively. The spatial dimensions were halved between the stages.

A global average pooling (GAP) layer is used after the final stage to reduce the spatial dimensionality. Therefore, the production of the GAP layer has a dimension of 1 × 1 × C, by taking the average per map. There is a softmax layer after the GAP layer. The filter size of all the convolutional layers except the last one is 3 × 3. The filter size of the last convolutional layer is 1 × 1. The GAP layer was proved successful in image processing applications [19]. It reduces the number of parameters of the models. For example, instead of using the 3 × 3 filter, the GAP uses the 1 × 1 filter, which decreases the number of parameters by one-ninth for subsequent operations.

3.3. Database

In this section, we discuss databases that were used in the experiments. Three databases were utilized in the experiments for the testing. The databases are ORL, FEI, and LFW. The ORL database has 10 images per subject, and there are 40 subjects [41]. So, the number of images is 400. The face images vary in angular taking, illumination, and facial expressions. Almost all the faces are having an upright frontal view, sometimes with a slight left or right rotation. The database was created at Cambridge University. The size of the images is 92 × 112.

FEI database is larger than the ORL database. The FEI database includes face images of 100 males and 100 females [42]. Each subject has 14 images; so, there are 2800 face images. The images have a size of 640 × 480. The faces are either frontal or angular and have expressions. Images are colorful and faces were taken in front of a uniform background.

Labeled Faces in the wild (LFW) is a big database that was designed to recognize faces in unconstrained environments [43]. There are 13 233 images of 5749 people, where 1680 people have two or more images. The size of the images is 250 × 250.

The proposed tree-based deep models were trained using the Canadian Institute for Advanced Research (CIFAR-10) database [40]. The training policy was adopted from [43]. Only flipping and translation were done according to [43], and no further augmentation was applied. The size of the images is 32 × 32.

4. Experimental results and discussion

Accuracy and information density were used to evaluate the models in the experiments. The trained models were fine-tuned by a small subset of a corresponding face database. In the case of the ORL database, three images per subject were selected for the fine-tuning; in the case of the FEI database, four images per subject were selected for the fine-tuning. As the LFW database does not much repetition of a subject's face image, only one image per subject was selected in the fine-tuning. The test was performed by the remaining images per subject. Therefore, the fine-tuning and test images were mutually exclusive.

The deep models' parameters were optimized by using a mini-batch gradient descent algorithm. The momentum value was set to 0.9 and the entropy loss was used as the loss function. The batch size was 128 and the learning rate was 0.1 at the beginning and reduced to one-tenth of the previous value in every 20 epochs. The number of branch factor and tree height were varied, and fixed to four and three, respectively, because they provided the optimal results.

First, we report the performance of the deep models in terms of epochs. Normally, if we increase the number of epochs, the accuracy increases until a certain number of epochs; then, the accuracy either drops or remains the same. All the experiments were performed using three databases mentioned earlier.

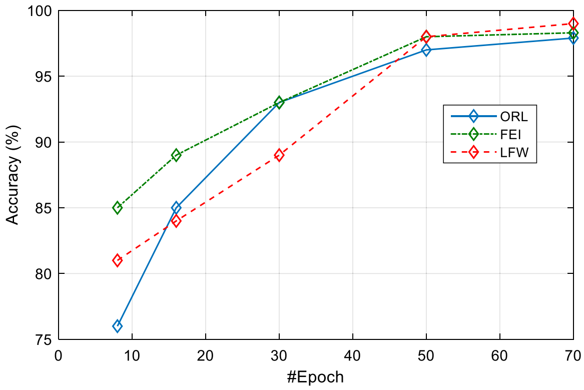

Fig. 4. Accuracy of the single tree model at different epochs using three databases.

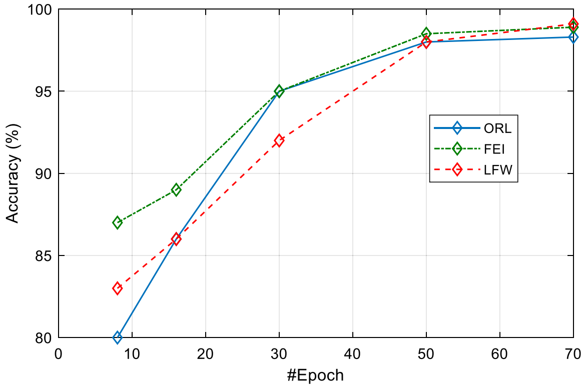

Fig. 5. Accuracy of the parallel tree model at different epochs using three databases.

Figs. 4 and 5 show the accuracies of the deep model using the single tree and the parallel trees, respectively, for different numbers of epochs. From both figures, we find the optimum number of the epoch was 50. At epoch 70, though accuracies slightly increased, it took more time than at epoch 50. In the case of real-time applications, time is an important factor. In subsequent experiments, we fixed the epoch number at 50.

Both models achieved similar accuracies in all the three databases at epoch 50. The accuracies ranged between 97% and 98.5%. The accuracy using the FEI database was better than using the other two databases.

Second, we report the performance of the models using different numbers of initial filters. If we have many filters, the number of parameters increases which increases the computational complexity of the system. Figs. 6 and 7 show the accuracies of the models using the single tree and the parallel trees, respectively. In both the models, the highest accuracies were obtained when the number of initial filters was 64. Please note that the number of filters was set to 16, 32, 64, or 128. When the number of filters was 128, the accuracies decreased slightly. The accuracy using the LFW database was better than using the other two databases.

The parallel tree model can be constructed using different numbers of tree modules per tree. If the number of modules is high, the system can run in parallel processors and the time is reduced to process. We investigated the effect of the number of modules per tree on accuracy using the three databases. Fig. 8 shows the accuracy of the model with a different number of modules. We varied the number of modules to 4, 6, 8, or 10. From the figure, we see that 8 modules or 10 modules achieved the best accuracy. For example, using 8 modules, the accuracy

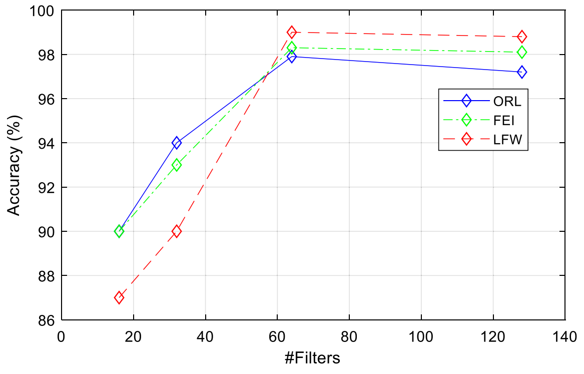

Fig. 6. Accuracy of the single tree model for different numbers of initial filters using three databases.

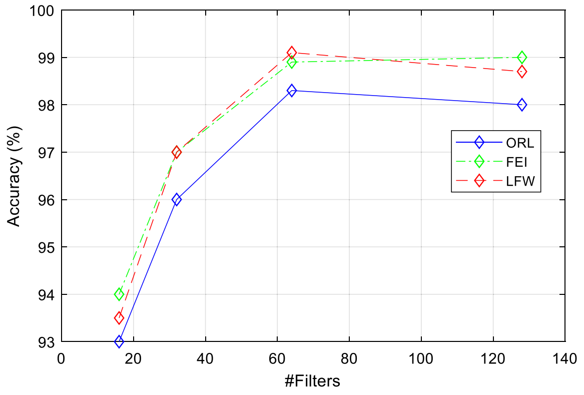

Fig. 7. Accuracy of the parallel tree model for different numbers of initial filters using three databases.

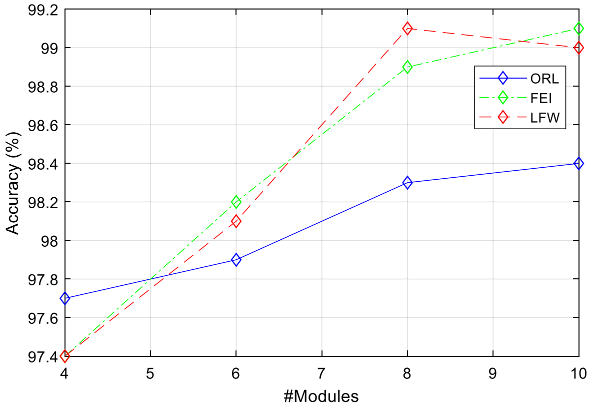

Fig. 8. Accuracy of the parallel tree model for different numbers of modules per tree using three databases.

was the highest for the LFW database. The accuracy slightly decreased using 10 modules for this database. For the other two databases, the accuracies were better at 10 modules than at 8 modules. The accuracy of using the ORL database was the least.

One of the main objectives of the proposed tree-based deep models is to achieve high accuracy in less time. A system performs processing in less time when the number of system parameters is less. The smart city applications need a system to operate in real-time with high accuracy. We investigated the proposed models in terms of the number of parameters and accuracy. A metric called the information density was used to evaluate the models as an indicator of the steadiness between the accuracy and the number of parameters. The information density is defined by the accuracy over the number of parameters in million. If the accuracy is high and the number of parameters is low, the information density is high which is good for a model. We compared different models in terms of information density. Fig. 9 shows the performance of different models. Four well-known deep models were taken into account for the comparison purpose. For this particular experiment, we changed the set up in a way that both the training and the testing were performed in the CIFAR-10 database. Therefore, the accuracies were not for face recognition but for image recognition; however, it does not affect the metric of information density of the models. The compared models are ResNext [44], wide residual network [9], VGG 16 [6], and MobileNet [45]. The VGG model is very deep and has high accuracy; however, it has many parameters. The MobileNet is not very deep and has a comparable accuracy. The ResNext and the wide residual network lie between the VGG model and the MobileNet in terms of depth and accuracy. Comparing these four models, the VGG net and the MobileNet models have much higher information density, while the other two models have less information density. The proposed tree-based deep models have higher information density than the four previous models. The single tree-based deep model outperformed all other models. For example, the proposed single tree-based model has an information density of 0.58, while the proposed parallel tree-based model has an information density of 0.28. The information densities of the VGG Net and the MobileNet are 0.26 and 0.25, respectively.

Fig. 10 shows the accuracies of face recognition with the LFW database using different deep neural network models. Here, we considered five existing models that achieved good accuracies and the proposed two models. From the figure, we find that the proposed parallel tree-based deep model gained the accuracy comparable with that by the Face Net [33]. The Face Net had an accuracy of 99.6%, while the proposed parallel tree-based model had an accuracy of 99.4%.

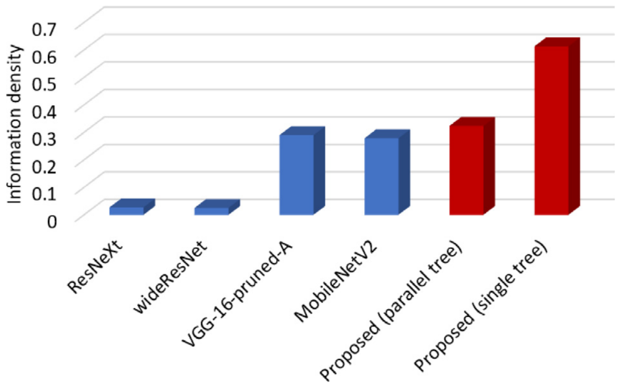

Fig. 9. The information density of the models.

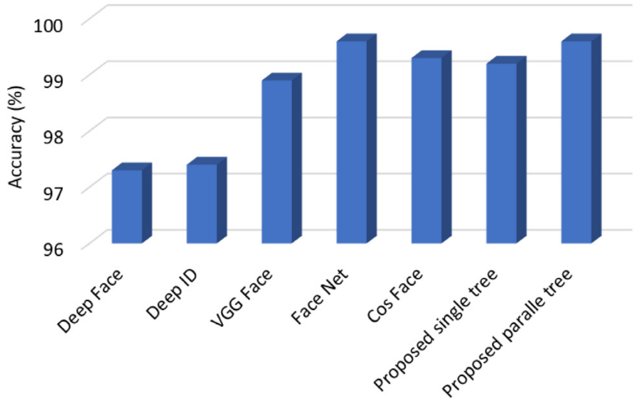

Fig. 10. Accuracy comparison between the models.

The Face Net is a very deep network that has many layers of convolution; therefore, the execution time is much higher than the proposed model. The Cos Face network also has a large number of parameters. Considering all these aspects we can say that the proposed tree-based deep model is very efficient for face recognition purposes.

The proposed models have branches, so the computation can be distributed to the branches to run in parallel. The operations in each branch are not dependent on the operations in other branches. Therefore, the computation time is less in the proposed model. The total numbers of parameters in the single tree-based deep model and in the parallel tree-based deep model are as follows.

$$O\left(n.c^2.\left(\frac{1}{b^L}+m^2.\left(\frac{1-b^L}{1-b}+\frac{1}{b^L}\right)\right)\right) \text{ For single tree-based model}$$

$$O\left(n.c^2.\left(b^L+m^2.\left(b^{L+1}-2-b\right)\right)\right) \text{ For parallel tree-based model}$$

where n is the number of filters having size $m \times m$, b and L are the branching factor and tree height, respectively, and c is the of channels of the image.

5. Conclusion

The tree-based deep models for face recognition were proposed in this paper. There were two versions of the proposed model: single tree and parallel trees. In the parallel trees model, several single trees were arranged in parallel. The residual function was used as the building block of the whole architecture. Experiments were performed using three publicly available face databases. The proposed models achieved around 99% accuracy using these databases. The information density of the proposed single tree-based model was near 0.6, which is considered excellent for a deep model. Compared to the existing deep models, the proposed models had comparable accuracies with a lesser number of parameters. These findings prove that the proposed models can be efficiently used as a face recognition system in real-time for security purposes. The future directions of the proposed work can be as follows. First, the proposed models can be extended to include more components such as the locally aggregated descriptors [46] instead of the concatenation used in the models. Second, more databases can be included in the experiments. Third, the proposed models can be used in other applications such as gender recognition and facial emotion recognition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Mehedi Masud: Investigation, Funding acquisition. **Ghulam Muhammad:** Conceptualization, Methodology, Writing - review & editing. **Hesham Alhumyani:** Formal analysis. **Sultan S Alshamrani:** Investigation. **Omar Cheikhrouhou:** Formal analysis. **Saleh Ibrahim:** Funding acquisition. **M. Shamim Hossain:** Supervision.

Acknowledgement

This study was funded by the Deanship of Scientific Research, Taif University, Saudi Arabia [Research project number 1-440-6146].

References

[1] M.S. Hossain, G. Muhammad, An audio-visual emotion recognition system using deep learning fusion for cognitive wireless framework, IEEE Wirel. Commun. Mag. 26 (3) (2019) 62–68.

[2] M.S. Hossain, G. Muhammad, Cloud-assisted industrial internet of things (IIoT) - enabled framework for health monitoring, Comput. Netw. 101 (2016) (2016) 192–202.

[3] L. Hou, et al., Internet of things cloud: Architecture and implementation, IEEE Commun. Mag. 54 (12) (2016) 32–39.

[4] Q. Fang, et al., Folksonomy-based visual ontology construction and its applications, IEEE Trans. Multimedia 18 (4) (2015) 702–713.

[5] C. Szegedy, et al., Going deeper with convolutions, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, 2015, pp. 1-9.

[6] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2014, arXiv preprint arXiv:1409.1556.

[7] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Adv. Neural Inf. Process. Syst. (2012) 1097–1105.

[8] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[9] S. Zagoruyko, N. Komodakis, Wide.residual. networks, Wide residual networks, 2016, arXiv preprint arXiv:1605.07146.

[10] M.S. Hossain, G. Muhammad, Emotion recognition using deep learning approach from audio-visual emotional big data, Inf. Fusion 49 (2019) 69–78.

[11] M.S. Hossain, G. Muhammad, Cloud-based collaborative media service framework for health-Care, Int. J. Distrib. Sens. Netw. (2014) 858712, 11 pages.

[12] A. Al-nasheri, et al., An investigation of multi-dimensional voice program parameters in three different databases for voice pathology detection and classification, J. Voice 31 (1) (2017) 113.e9–113.e18.

[13] M.S. Hossain, S.U. Amin, G. Muhammad, M. Al Sulaiman, Applying deep learning for epilepsy seizure detection and brain mapping visualization, ACM Trans. Multimedia Comput. Commun. Appl. 15 (1s) (2019) Article 10, 17 pages.

[14] M. Masud, et al., Data interoperability and multimedia content management in e-health systems, IEEE Trans. Inf. Technol. Biomed. 16 (6) (2012) 1015–1023.

[15] M.S. Hossain, M.A. Rahman, G. Muhammad, Cyber-physical cloud-oriented multi-sensory smart home framework for elderly people: An energy efficiency perspective, J. Parallel Distrib. Comput. 103 (2017) (2017) 11–21.

[16] M. Chen, et al., Secure enforcement in cognitive internet of vehicles, IEEE Internet Things J. 5 (2) (2018) 1242–1250.

[17] M.F. Alhamid, et al., Towards context-sensitive collaborative media recommender system, Multimed. Tools Appl. 74 (24) (2015) 11399–11428.

[18] G. Muhammad, M. Alsulaiman, S.U. Amin, A. Ghoneim, M.F. Alhamid, A facial-expression monitoring system for improved healthcare in smart cities, IEEE Access 5 (1) (2017) 10871–10881.

[19] A.A. Amory, G. Muhammad, H. Mathkour, Deep convolutional tree networks, Future Gener. Comput. Syst. 101 (2019) 152–168.

[20] X. Yang, et al., Deep relative attributes, IEEE Trans. Multimedia 18 (9) (2016) 1832–1842.

[21] G. Muhammad, M.S. Hossain, A. Yasmine, Tree-based deep networks for edge devices, IEEE Trans. Ind. Informat. 16 (3) (2020) 2022–2028.

[22] J. Yu, K. Sun, F. Gao, S. Zhu, Face biometric quality assessment via light CNN, Pattern Recognit. Lett. 107 (2018) 25–32.

[23] Y. Sun, X. Wang, X. Tang, Hybrid deep learning for computing face similarities, Int'l Conf. Comput. Vis. 38 (10) (2013) 1997–2009.

[24] R. Singh, H. Om, Newborn face recognition using deep convolutional neural network, Multimedia Tools Appl. 76 (18) (2017) 19005–19015.

[25] K. Guo, S. Wu, Y. Xu, Face recognition using both visible light image and near-infrared image and a deep network, CAAI Trans. Intell. Technol. 2 (1) (2017) 39–47.

[26] H. Hu, S.A.A. Shah, M. Bennamoun, M. Molton, 2D and 3D face recognition using convolutional neural network, in: TENCON 2017-2017 IEEE Region 10 Conference, Penang, 2017, pp. 133-132.

[27] M. Simón, et al., Improved RGB-d-t based face recognition, IET Biometr. (2016) 297–304.

[28] Z. Lu, X. Jiang, A. Kot, Deep coupled resnet for low-resolution face recognition, IEEE Signal Process. Lett. 25 (4) (2018) 526–530.

[29] Y. Taigman, M. Yang, M. Ranzato, L. Wolf, Deepface: Closing the gap to human-level performance in face verification, in: Proc. IEEE Conf. Comput. Vis. pattern Recognit. 2014, pp. 1701–1708.

[30] Y. Sun, X. Wang, X. Tang, Deep Learning Face Representation from Predicting 10, 000 Classes, in: Proc. IEEE Conf. Comput. Vis. pattern Recognit. 2014, pp. 1891–1898.

[31] Y. Sun, X. Wang, X. Tang, Deeply learned face representations are sparse, selective, and robust, in: IEEE Conf. Comput. Vis. Pattern Recognit, 2015, pp. 2892–2900.

[32] O.M. Parkhi, A. Vedaldi, A. Zisserman, et al., Deep face recognition, in: BMVC, Vol. 1, 2015, p. 6.

[33] F. Schroff, D. Kalenichenko, J. Philbin, Facenet: A unified embedding for face recognition and clustering, in: CVPR, 2015, pp. 815–823.

[34] F. Wang, W. Liu, H. Liu, J. Cheng, Additive margin softmax for face verification, 2018, arXiv preprint arXiv:1801.05599.

[35] H. Wang, Y. Wang, Z. Zhou, X. Ji, Z. Li, D. Gong, J. Zhou, W. Liu, Cosface: Large margin cosine loss for deep face recognition, 2018, arXiv preprint arXiv:1801.09414.

[36] A. Yassine, et al., IoT big data analytics for smart homes with fog and cloud computing, Future Gener. Comput. Syst. 91 (2019) 563–573.

[37] G. Muhammad, M.F. Alhamid, M. Alsulaiman, B. Gupta, Edge computing with cloud for voice disorders assessment and treatment, IEEE Commun. Mag. 56 (4) (2018) 60–65.

[38] J. Wang, et al., A software defined network routing in wireless multihop network, J. Netw. Comput. Appl. 85 (2017) (2017) 76–83.

[39] Y. Zhang, et al., Edge intelligence in the cognitive internet of things: Improving sensitivity and interactivity, IEEE Netw. 33 (3) (2019) 58–64.

[40] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[41] ORL face database. [Online]. Available: http://www.uk.research.att.com/facedatabase.html.

[42] C.E. Thomaz, FEI Face database, 2012, [Online]. Available: https://fei.edu.br/~cet/facedatabase.html.

[43] G.B. Huang, M. Ramesh, T. Berg, E. Learned-Miller, Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments, Technical Report 07-49, University of Massachusetts, Amherst, 2007.

[44] S. Xie, R. Girshick, P. DolláR, Z. Tu, K. He, Aggregated residual transformations for deep neural networks, in: Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on, 2017, pp. 5987–5995.

[45] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, Mobilenetv2: Inverted residuals and linear bottlenecks, 2018, arXiv preprint arXiv:1801.04381.

[46] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, J. Sivic, NetVLAD: CNN architecture for weakly supervised place recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 5297-5307.