Emotion Recognition for Cognitive Edge Computing Using Deep Learning

Ghulam Muhammad, *Senior Member, IEEE*, and M. Shamim Hossain, *Senior Member, IEEE*

***Abstract*—The growing use of the Internet of Things (IoT) has increased the volume of data to be processed by manifolds. Edge computing can lessen the load of transmitting a massive volume of data to the cloud. It can also provide reduced latency and real-time experience to the users. This article proposes an emotion recognition system from facial images based on edge computing. A convolutional neural network (CNN) model is proposed to recognize emotion. The model is trained in a cloud during off time and downloaded to an edge server. During the testing, an end device such as a smartphone captures a face image and does some preprocessing, which includes face detection, face cropping, contrast enhancement, and image resizing. The preprocessed image is then sent to the edge server. The edge server runs the CNN model and infers a decision on emotion. The decision is then transmitted back to the smartphone. Two data sets, JAFFE and extended Cohn–Kanade (CK+), are used for the evaluation. Experimental results show that the proposed system is energy efficient, has less learnable parameters, and good recognition accuracy. The accuracies using the JAFFE and CK+ data sets are 93.5% and 96.6%, respectively.**

***Index Terms*—Deep learning, edge computing, emotion recognition, Internet of Things (IoT).**

I. Introduction

EGDE computing has been introduced mainly to decrease the transmission load of data to the cloud. In this age of numerous sensors and the Internet of Things (IoT), the volume of data is increasing exponentially. A huge volume of data needs a high computing power to process, which is not available to end devices, such as smartphones and PDAs. Cloud computing can host workstations or supercomputers and provide large data storage. Therefore, the computation and storage of a huge volume of data can be done in cloud computing. The problem is the transmission of this volume of data. It needs enough bandwidth and sufficient time which may not be available due to interruption. Therefore, for a seamless transmission, the data volume should be sufficiently reduced, and if possible, some processing can be done before transmission. Edge computing can bring a solution to this [1]. Nowadays, powerful devices can be installed as an edge server, so edge computing can do a lot of processing before transmitting data to the cloud. For industrial applications, where real-time solutions are needed, powerful edge computing coupled with 5G technology renders more appeal to the consumers [2].

Deep learning is used in many applications giving high accuracies. It eliminates the need for handcrafted features. A deep neural network (DNN) is computationally expensive because it involves many matrix multiplications. A DNN consists of many layers, which are arranged in a sequence. A special kind of DNN is a convolutional neural network (CNN), which has been effectively used in many speech processing, image processing, video processing, and analysis applications. Deep learning requires a lot of computations; hence, a powerful machine is needed to train a DL model. A communal tactic is to leverage edge and cloud computing. In this case, the user data should be transferred from the input sensors or IoT to edge computing or cloud computing. While offloading the data from the sensors or IoT, there are three major challenges. The challenges are latency, scalability, and security.

Low latency is required for a real-time application, such as emotion recognition, traffic situation, speech recognition, and object detection. For example, cognitive computing needs to classify an emotion in real time from a video to use this emotion for a classified application. In this case, if the emotion is not classified in real time, the subsequent processing, which is dependent on the emotion will be delayed and can be meaningless. For an emotion inference, sending video data from the sensor to the cloud, classification in the cloud, and return the decision incurs many delays, which is an obstacle to an end-to-end (e2e) latency requirement.

Nowadays, the number of sensors and IoT devices has been increased significantly. Therefore, scaling data is needed to deal properly with network resource utilization. If there is no scaling, the network access to the cloud will be a bottleneck. Especially, if the data are of massive volume such as video, the bandwidth is a concern.

Edge computing can be a solution to meet the criteria of low latency, scalability, and data security. The edge server can be located within the proximity of a base station and sensors or IoT gateways [3]. Edge devices consist of an edge server, a cellular base station, and several end devices (data sources), such as smartphones, Raspberry Pi, wireless sensors with IP, and IoT. The proximity of the edge server to the data sources reduces the e2e latency and provides a real-time solution.

Manuscript received September 7, 2020; revised December 4, 2020 and January 27, 2021; accepted February 8, 2021. Date of publication February 10, 2021; date of current version November 19, 2021. This work was supported by the Deanship of Scientific Research at King Saud University, Riyadh, Saudi Arabia, through the Vice Deanship of Scientific Research Chairs: Chair of Pervasive and Mobile Computing. *(Corresponding author: M. Shamim Hossain.)*

Ghulam Muhammad is with the Department of Computer Engineering, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia (e-mail: ghulam@ksu.edu.sa).

M. Shamim Hossain is with the Chair of Pervasive and Mobile Computing, and also with the Department of Software Engineering, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia (e-mail: mshossain@ksu.edu.sa).

Digital Object Identifier 10.1109/JIOT.2021.3058587

Edge computing with the help of a 5G network can provide solutions to latency, scalability, and security. The challenges remain in the edge computing to run deep learning. Deep learning needs highly resourced computing devices to train; however, the edge devices have a variety of resource power, for example, GPU-equipped edge servers, smartphones with mobile processors, and medium-resourced Raspberry Pi devices. While deep learning processing can be distributed over different devices, the coordination and distribution of various devices is a potential challenge. The devices have heterogeneous processing capabilities, and are available at various times, making them very difficult to coordinate between themselves. A good scheduling and optimization algorithm can mitigate this problem.

This article proposes a deep-learning-based system to recognize emotion from faces using edge computing. Recently, research on cognitive computing is flourishing due to its many interesting applications, such as smart healthcare [36], smart city, smart advertisement, and smart living [4]. Human performances on cognitive tasks, such as speech or speaker recognition and image classification are very accurate and fast. For instance, face recognition takes only a couple of hundreds of milliseconds (ms), human voice identification takes no more than 10 ms, and speech recognition of a short phrase takes around the same time as face recognition [5].

In the proposed deep-learning-based system, smart sensors such as smartphones capture facial frames, do some preprocessing, and transmit them to the edge server via a low-range wireless networks, such as Bluetooth or WiFi. The edge server runs a pretrained deep learning model on the frames, provides a decision on the emotion classification, and sends the decision to smartphones. The communication between smartphones and the edge server is done using 5G to reduce latency and increase consumer satisfaction. A cloud server is also used to train the deep learning model and store face data. The communication between the edge server and the cloud server is done during the off-period, and therefore, does not affect the latency.

The contribution of this article can be stated as: 1) introducing a facial emotion recognition system leveraging edge computing; 2) reducing the latency and improving consumer satisfaction by applying preprocessing in end devices, testing in the edge server, and training in the cloud; and 3) evaluating the system using multiple data sets.

This article is organized in the following manner. Section II delivers a literature review on facial emotion recognition, Section III presents the proposed system, Section IV gives evaluation setup, results, including recognition accuracy and energy consumption, and Section V provides the conclusion.

II. Literature Review

There exist many emotion recognition systems in the literature. Early systems utilized mainly handcrafted features and classical classifiers [6]. Pioneering work on emotion recognition was by Paul Ekman, who recognized six emotion categories: 1) anger; 2) disgust; 3) fear; 4) happiness; 5) sadness; and 6) surprise, in addition to neutral [7]. These emotions are called the seven basic emotions, which are adopted by most of the emotion data sets.

There are several publicly available emotion data sets to develop emotion recognition systems. Two facial emotion data sets are JAFFE [8] and the extended Cohn–Kanade (CK+) data set [9].

Since the development of deep learning models, they have been effectively used in numerous applications, including image processing, image and object recognition, speech and speaker applications, and medical signal processing [10]–[12]. For image processing and recognition applications, a CNN is widely used. In this article, we mainly focus on emotion recognition systems using deep learning models or CNN models.

A transfer learning of the CNN model trained on image classification task was used to recognize facial emotions in [13]. The authors achieved 55.6% accuracy using the EmotiW data set. A deep sparse autoencoder was used using the Histogram of Oriented Gradients (HoG) features in [14]. An accuracy of 96% was attained using the CK+ data set.

Using the same CK+ data set, several other works obtained different accuracies. For example, a DNN was used to raw face images in [15] and got 93.2% accuracy. A zero-bias CNN was applied to emotion recognition and got a high accuracy [16]. A boosted deep belief network was utilized by combining the feature extraction part and the classifier part in one loop and found to be efficient in the CK+ and the JAFFE data sets [17]. A deep model named Face Net2ExpNet by combining different architectural CNNs was proposed in [18] and obtained an accuracy of 96.8%.

A relativity learning-based DNN was proposed to recognize emotion from faces in [19], where an accuracy of more than 70.0% was obtained in the FER-2013 data set. A 3DCNN-based system was developed by taking into consideration the facial action parts and learned parts in [20]. Emotion can also be used for patient monitoring [21].

There are some recent advances in emotion recognition using CNN. Jain *et al.* proposed a residual CNN model in [22]. Some normalization in the form of contrast enhancement was done before the model. The authors achieved 95.2% accuracy on the JAFFE data set and 93.2% accuracy on the CK+ data set. Two different 3-D convolutional networks were used together with action units' enhancement in [23]. The two networks were 3DLeNet and EvoNet, and the fusion was done by an adaptive subsequence matching algorithm. Two-level attention and two-stage multitask learning were used in [24] to recognize emotion. In this method, attention was given to some corresponding regions of the face. Fernandez *et al.* [25] proposed a CNN model consisting of four modules: 1) attention; 2) feature extraction; 3) reconstruction; and 4) classification. The model achieved 90.3% accuracy using the CK+ data set. In [26], an attention CNN was proposed using a feature extraction network and a localization network. It obtained 92.8% and 98.0% accuracy using the JAFFE and the CK+ data sets, respectively.

A hybrid label-less learning, which can automatically label the data without human involvement, was introduced by using the model of similarity and the entropy in [34]. A multimodal

and multiscenario simulation approach based on an attention evaluation was proposed in [35]. The relationship between emotional evidence and recognition was also carefully analyzed and the labels were corrected.

Though there are many emotion recognition systems, only a few had been using edge computing [4], [27], [33]. However, these systems did not fully utilize the computation of the edge. Recently, many applications are using edge computing to offload the computation to the edge to take the advantage of edge processing power and reduce the burden to transmit data to the cloud [28]–[30]. Powered by the edge computing concept and the progress of artificial intelligence-based, rich cognitive resources, a new computing model, edge cognitive computing (ECC), is a groundbreaking method that extends cognitive computing at the edge of the network. Amin *et al.* [31] introduced a complex application transformation model based on the electroencephalograph (EEG) signals by providing a perspective into how cognitive computing and artificial intelligence merge.

In [32], a smart-healthcare method was introduced, based on the ECC. Using cognitive computation, this method was able to track and assess users' physical fitness. It also adequately changed the computational resource distribution for the entire edge computing network for each user's health-risk ranking. In [39], to explore the viability of the cognitive IoT–cloud smart healthcare platform [41], [42], an EEG seizure prediction approach was suggested using deep learning. The authors employed smart EEG sensors in the proposed system to monitor and pass on EEG signals from epileptic patients. The cognitive system subsequently made a real-time judgment on future events and whether to forward the data to the deep learning node [37], [38]. In this article, we develop a CNN-based emotion recognition system by utilizing the advantages of edge computing.

III. Proposed System

In this section, the proposed emotion recognition system is described.

A. Edge Computing Structure

Fig. 1 shows the edge computing structure of the proposed system. Edge devices are of two types: 1) end devices (IoT) and 2) the edge server. End devices include smartphones, Raspberry Pi, IP cameras, etc. Face images are captured by an end device, for example, a smartphone. The images are represented by a frames' sequence. The smartphone does some preprocessing on images and transfers the preprocessed images to the edge server using the 5G network. The edge server is in the proximity of the end device. The server is equipped with a GPU and good computing power. The cloud, which we mention as the core cloud in the figure, hosts a global deep model for emotion recognition. The cloud has also global storage. The global storage stores all the face images acquired by different devices from various places. These images are the training data of the global deep model. The proposed deep model is trained in the cloud and is stored as the global deep model.

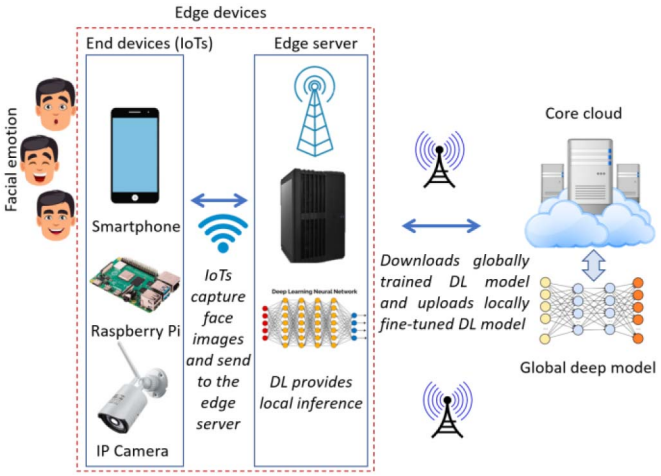

Fig. 1. Framework of edge computing in the proposed system.

The edge server has two tasks. During off time, the edge server downloads the global deep model from the cloud. When the smartphone sends any face image to the edge server, it executes the deep model (inferring) and gives a decision, which is then passed to the smartphone. The inferring is a continuous process because the image frames are coming continuously.

A time division analysis of edge computing can be formularized as follows. It can be mentioned that the main objective is to get accurate emotion recognition and to reduce the latency provided that the end devices have no sufficient power to execute a deep model. The whole time required to get a decision in the end device for an image can be divided into two major time durations: 1) image preprocessing (in the end device) and 2) offloading. Image preprocessing time includes the time required for face detection and cropping, contrast enhancement, and resizing. Offloading time includes data transfer time from the end device to the edge server, infer time in the server, and the decision transfer time from the edge server to the end device. Fig. 2 shows the illustration of the time required to and from the edge server.

The equation to determine different times is shown below with timestamps T

$$\begin{aligned} t_{\text{preprocess}} &= T_2 - T_1 \\ t_{\text{frame\_transfer}} &= T_3 - T_2 \\ t_{\text{process\_server}} &= T_4 - T_3 \\ t_{\text{decision\_transfer}} &= T_5 - T_4 \\ t_{e2e} &= \begin{cases} t_{\text{preprocess}} + t_{\text{frame\_transfer}} \\ + \ t_{\text{process\_server}} + t_{\text{decision\_transfer}} \end{cases} \\ t_{\text{offload}} &= \begin{cases} t_{\text{frame\_transfer}} + t_{\text{process\_server}} \\ + \ t_{\text{decision\_transfer}}. \end{cases} \end{aligned} \tag{1}$$

The latency is the time between T_5 and T_1. In this case, the cloud processing time and the time required for the transmission between the edge and the cloud are not considered, because they are done during off time.

In the proposed system, there are three basic components and two optional components. Table I lists the components and corresponding tasks. The optional components involve the cloud. The three basic components are 1) the end-device

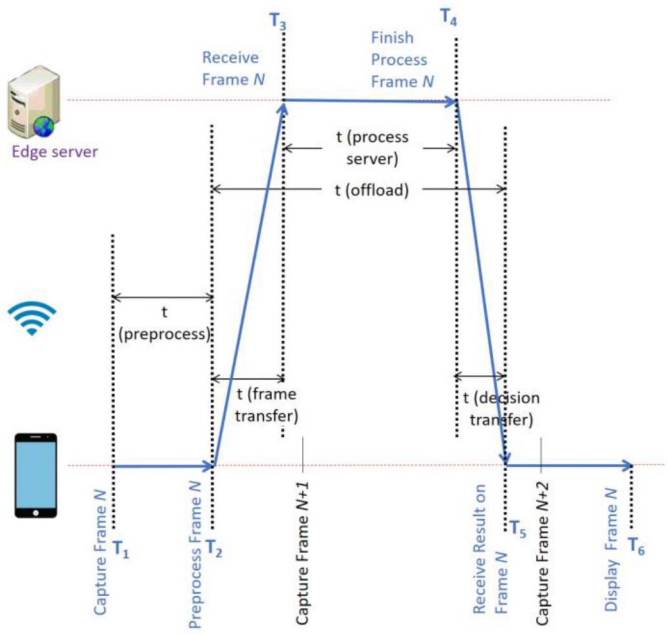

Fig. 2. Time requirement illustration to and from the end device and the edge server.

TABLE I
TASKS OF DIFFERENT COMPONENTS IN THE PROPOSED SYSTEM

Component name	Tasks
End device component	▪ Face detection ▪ Cropping ▪ Contrast enhancement ▪ Resizing
Communication component	▪ Transmit frames ▪ Convey decision (result)
Edge server component	▪ Update DL model ▪ Find emotion class ▪ Visualize (optional)
Cloud server component	▪ Update the global model using data from different edge servers located at various places
Communication component - cloud	▪ Download global DL model to the edge ▪ Upload the updated DL model to the cloud

component; 2) the communication component; and 3) the edge server component. The edge device component elements are face detection, face cropping, contrast enhancement, and image resizing; these elements are collectively called preprocessing. Face detection is used by the face detection module of the smartphone. The rectangular area, in which the face is detected, is cropped. Then, a contrast enhancement algorithm based on the Gamma-law is used. If the image is color, it is converted to grayscale. Then, the image is resized to 227 × 227, which is the requirement of the proposed CNN model. The edge device component elements ensure that the image is only containing the face and the size of the data is small so that the transmission to the edge server does not take much bandwidth or time.

In the communication component, there are two elements. The first one is the transmission of the image from the end device to the edge server, which occurs just after the preprocessing. The second one is to convey the decision (the class of the emotion) from the edge server to the end device, which happens after the inferring in the edge server.

In the edge server component, there are three tasks. When an image comes from the edge device, the server runs the DL model (in our case, the CNN model) and infers a decision. It may also provide a visualization of the active neurons of the model if needed. If the edge server gathers enough images, it uses them to update the DL model by fine-tuning during off time.

In the proposed system, the cloud is not used for inferring; however, it is used mainly to train the model using a massive volume of data. In the cloud communication component, there are two transfers, one to download the global DL model from the cloud to the edge server, and the other to upload the updated DL model from the edge server to the cloud. The cloud server component first trains the DL model using all the available data to produce the global DL model. When it receives more data from different sources, it updates the DL model. The training and updating of the global DL model occur offline and have no contribution to the latency calculation of the proposed system.

B. Proposed CNN Model for Emotion Recognition

Many CNN models in the literature perform excellently in many applications. Most of these models are trained with millions of samples, and computationally dense. They have many layers and a huge number of learnable weights. These models, though excellent in accuracy, are not suitable in devices with low computation power, such as smartphones or the edge server. Therefore, a new light CNN model is developed to recognize emotion from face images.

Fig. 3 shows the block diagram of the proposed CNN model. The model takes advantage of connection networks. The model has four main blocks of convolution. Each block has two convolution layers, and one shortcut connection via 1×1 convolution, and one max-pooling layer, as shown in the figure. The convolution layers use 3×3 convolutions with stride 2 in the first block and stride 1 in the following blocks, and there is padding before each convolution layer. Each convolution layer and the connection layer are followed by a rectifier linear (ReLU) unit. The stride is 2 for the max-pooling layer.

After the fourth block, there is a global average pooling (GAP), followed by a 50% dropout, two fully connected (FC) layers, and a softmax layer. The first FC layer has 2048 neurons, whereas the second one has 1024 neurons. The number of filters in each convolution is shown in each rectangular box of convolution in the figure. The input image size is 227×227. The output size of the first convolution layer is, therefore, $114 \times 114 \times 16$.

The training was done using class cross entropy as the error cost function with a mini-batch size of five samples. The face image samples were augmented using a scaling factor between 0.8 and 1.2, and rotation between $-25°$ and $25°$ angles. The

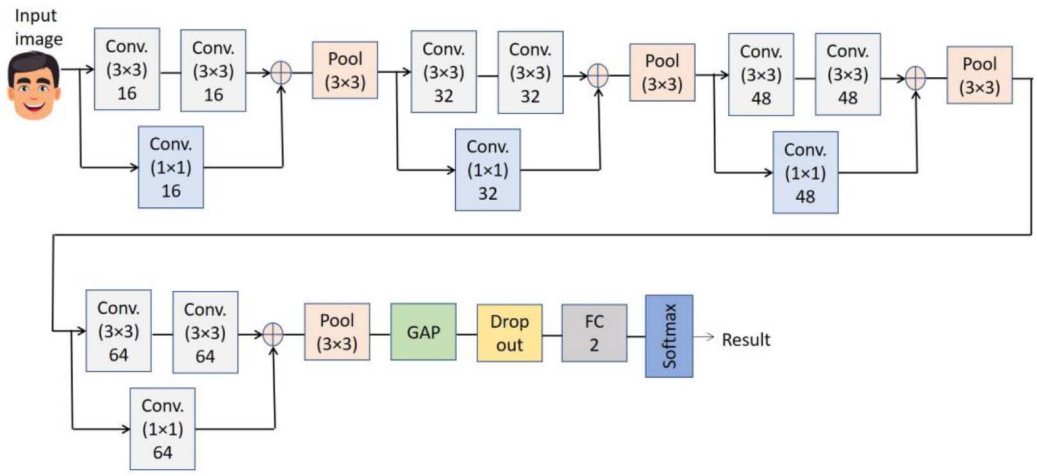

Fig. 3. Proposed CNN model for emotion recognition.

weights of the filters were initialized randomly using a zero mean and one standard deviation. The Adam optimizer was used to update the weights. The parameter values of the Adam optimizer were set to $\beta_1 = 0.9$, $\beta_2 = 0.999$, epsilon $= 1 \times 10^{-8}$, and the learning rate $= 5 \times 10^{-5}$. All the parameters were set empirically.

The proposed model has only about 190 000 learnable weights or parameters, which is very few compared to other existing models. For example, GoogleNet has 7.0M parameters (M stands for million), mobilenetV2 has 3.5M, ResNet 18 has 11.7M, ResNet101 has 44.6M, AlexNet has 61M, and InceptionV3 has 23.9M parameters according to MATLAB user guide. The only lighter model than the proposed model is squeezenet, which has 1.24M parameters; however, squeezenet did not perform well in our experiments. For a less or moderate size of input data, a compact model with few degrees of freedom avoids overfitting and can generalize well. Besides, a compact model with fewer parameters can easily be run on the edge server that may have a restricted processing power.

IV. Experiments

This section describes the data sets, implementation details, experimental protocol, and evaluation results with discussion.

A. Data Sets

Two publicly available data sets were used in the experiments. These data sets are JAFFE and CK+.

The JAFFE data set contains facial images of ten Japanese female actresses [8]. There is a total of 213 images of seven emotion categories (including neutral). Each actress provided two to four poses per emotion category. The image size is 256×256.

The CK+ contains facial emotion video samples of 123 people [9]. There is a total of 593 samples of seven basic emotion categories (there is no neutral): the categories are angry, contempt, disgust, fear, happy, sad, and surprise. The people were not trained before the recording. The videos have varied frame rates ranging between 10 and 60 frames/s. The image size is either 640×490 or 640×480. The recording was in grayscale and equal lighting conditions.

B. Implementation

A prototype is developed to evaluate the efficiency and accuracy of the proposed system. The prototype has three components: 1) an end device component, which is implemented on Android version 10; 2) an edge server component, which is realized using CUDA 10.0-enabled NVIDIA GeForce RTX 2070 8-GB GPU drivers, cuDNN v7.6 for deep learning models, and TensorFlow 2.0; and 3) a communication component, which has two parts, one is running in the smartphone using Apache HttpClient to communicate with the server and the other is running in the server using Django.

For the front-end component, we develop an interface in a Samsung Galaxy S20 Plus, 128-GB smartphone running on Android version 10. Some image preprocessing algorithms, such as contrast enhancement using Gamma-law, resizing to 227×227, and cropping are implemented as an App in the interface. These algorithms are developed using OpenCV libraries. More specifically, we use Android Studio 3.2, in which Java script can easily run. We integrate Dlib 19.16 and OpenCV 4.0.1 using LLDB, CMake, and NDK tools.

C. Protocol

The system was validated using a fivefold cross-validation tactic, where all the samples were randomly divided into equal fivefold and in each repetition, fourfold were used in the training and the other fold in the testing. Therefore, after five repetitions, all the samples were tested. The results reported here are the average results of the iterations. The training was done in a cloud server, then the trained model was downloaded on the edge server. The image preprocessing of the test images was done in the smartphone, and the testing using the trained CNN model was done in the edge server.

D. Results and Discussion

Figs. 4 and 5 display the confusion matrices of the proposed system using the CK+ and the JAFFE data sets, respectively.

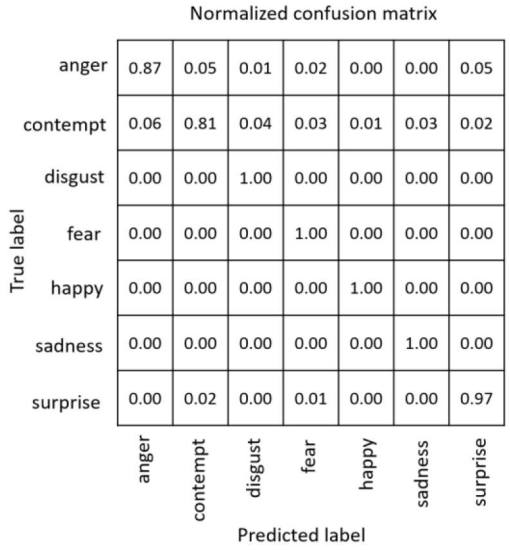

Fig. 4. Confusion matrix of the system using the CK+ data set.

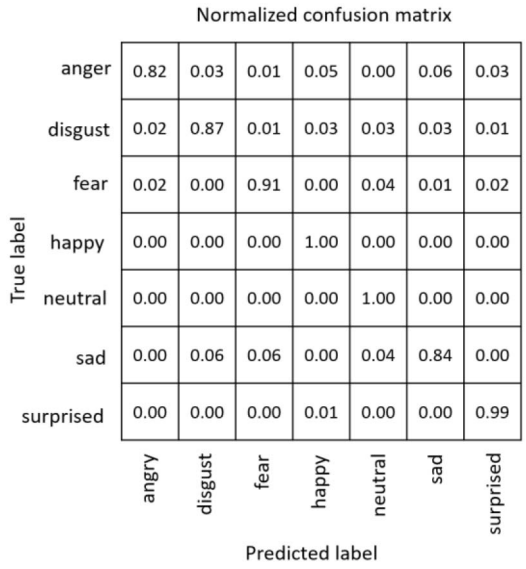

Fig. 5. Confusion matrix of the system using the JAFFE data set.

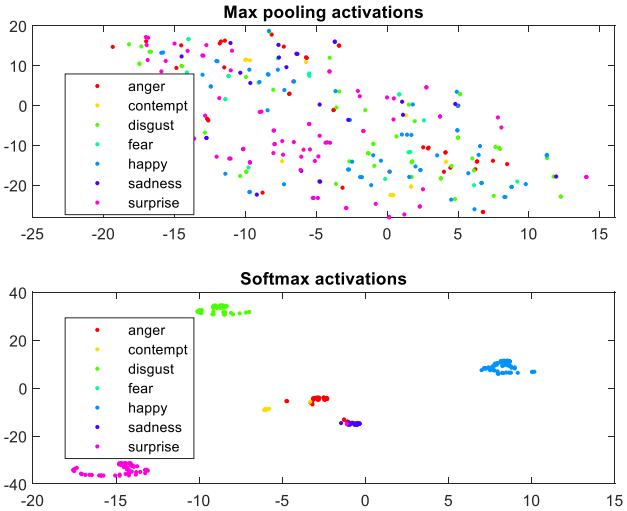

Fig. 6. Illustration of class separation capability of the proposed system. For details, see the text.

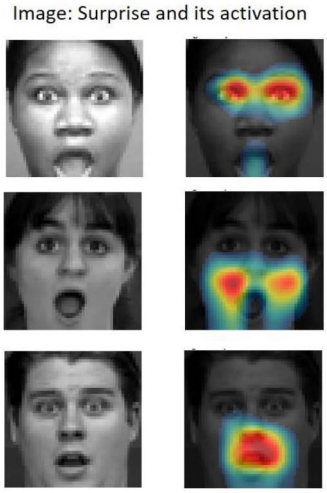

Fig. 7. Activation areas of different surprise face images.

There are seven emotion categories in the CK+ data set. Disgust, fear, happy, and sadness got 100% accuracy, while anger and contempt had below 90% accuracy. The anger class was mostly classified as the sadness class. Also, some contempt samples were misclassified as the fear class. The overall accuracy was 96.6%. The JAFFE data set has seven emotion categories including the normal. Happy and neutral categories had 100% accuracy, followed by the surprised category having 98.7% accuracy. The angry class was confused with the sad class, while some sad samples were classified as the happy class. The overall accuracy was 93.5%.

Fig. 6 illustrates the class separation capability of the proposed system by using the *t* distribution-stochastic neighborhood embedding (t-SNE) method [40]. The t-SNE is a strategy to transform a collection of high-dimensional vectors into a collection of lower dimensional ones while preserving the relative resemblance of classes as similar as possible to the original. The validation samples of different categories are plotted after the first max-pooling activations (upper figure) and after the softmax (lower figure). The figure is drawn using the CK+ data set. From this figure, we infer that the proposed system can effectively cluster the samples according to the categories.

Fig. 7 shows the activation areas of different surprise faces. From the figure, we see that for the same emotion category, the activation areas may vary, but mainly concentrate on either eyes, cheeks, or mouth areas. In future work, we may need to work on these attention areas to improve performance.

We compared four different systems for energy consumption in Joules. The first system consists of a local binary pattern feature extraction module and a support vector machine classifier, both running on the smartphone, and we name the system as the classical system. The second system is like the first system except it uses CNN instead of handcrafted features. The third system employs the proposed CNN model running at the edge server with preprocessing; we name it the edge

TABLE II
ENERGY CONSUMPTION (JOULES) OF THE SYSTEMS PER IMAGE

System	Image analysis	Image transferring
Classical system	1.05	0
Classical system with CNN	1.51	0
Edge system	0	0.92
Proposed system	0.55	0.58

TABLE III
COMPARISON OF ACCURACIES (%) OF DIFFERENT SYSTEMS IN THE LITERATURE

System	JAFFE	CK+
[4]	-	98.90
[16]	82.43	81.48
[22]	95.23	93.24
[25]	-	90.30
[26]	92.80	98.00
Proposed	93.50	96.60

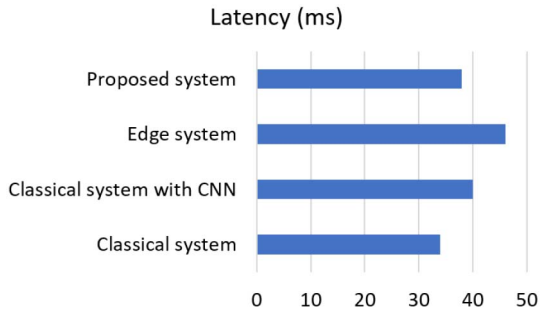

Fig. 8. Latency comparison between the systems.

system. The fourth one is the proposed system where the preprocessing is at the smartphone and the CNN model running at the edge.

Table II shows the energy consumption of these three systems. The proposed system had comparable energy consumption with the other two systems. For example, the proposed system consumed 1.13-J energy, which was comparable. It can be noted that the accuracy of the first system was below 75% using the CK+ data set, and that of the third system was 86.2%.

Table III shows the comparison of accuracies between different state-of-the-art systems and the proposed system. All the compared systems are very recent, published in 2019 or 2020. The accuracies of these systems were taken from the corresponding papers. We also performed experiments with the ResNet50, ResNet101, and Inception v3 models. Using the CK+ data set, the accuracies of these models were 97.7%, 97.1%, and 95.4%, respectively. It may be noted that all but the proposed system has learnable parameters of more than 10M, which is a big concern in our approach. Despite the proposed system has very few parameters, it achieved comparable accuracy with other systems.

The latency of the system can be reduced by 1) reducing the resolution of the input face image using compressive sampling; 2) encoding bitrate; and 3) incorporating an adaptive offloading of frames. Fig. 8 shows the latency comparison between the four systems. From the figure, we understand that the proposed system almost had the same latency as the classical system, which runs on the smartphone (local). The proposed system had a much lower latency than that of the edge system.

V. CONCLUSION

An edge-centric emotion recognition system was realized in this article. A light CNN model was developed into the system. In the system, an end device such as a smartphone captured a face image and did some preprocessing to make the image compact and information-rich. The preprocessed image was sent to the edge server and the server inferred a decision using the CNN model. Experimental results proved that the proposed system was energy efficient. The system achieved 92.20% and 96.60% accuracy using the JAFFE and the CK+ data sets. The time needed to get a decision after capturing the face image was on the average 100 ms, which is acceptable for a real-time application.

In a future study, we will investigate the feasibility of making the CNN model even lighter so that it can be run on a smartphone.

REFERENCES

[1] M. Chen, Y. Hao, L. Hu, M. S. Hossain, and A. Ghoneim, "Edge-CoCaCo: Toward joint optimization of computation, caching, and communication on edge cloud," *IEEE Wireless Commun.*, vol. 25, no. 3, pp. 21–27, Jun. 2018.
[2] G. Muhammad, M. S. Hossain, and A. Yassine, "Tree-based deep networks for edge devices," *IEEE Trans. Ind. Informat.*, vol. 16, no. 3, pp. 2022–2028, Mar. 2020.
[3] Y. Zhang, X. Ma, J. Zhang, M. S. Hossain, G. Muhammad, and S. U. Amin, "Edge intelligence in the cognitive Internet of Things: Improving sensitivity and interactivity," *IEEE Netw.*, vol. 33, no. 3, pp. 58–64, May/Jun. 2019.
[4] M. S. Hossain and G. Muhammad, "Emotion recognition using secure edge and cloud computing," *Inf. Sci.*, vol. 504, pp. 589–601, Dec. 2019.
[5] T. R. Agus, C. Suied, S. J. Thorpe, and D. Pressnitzer, "Characteristics of human voice processing," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Paris, France, 2010, pp. 509–512.
[6] G. Muhammad and M. F. Alhamid, "User emotion recognition from a larger pool of social network data using active learning," *Multimedia Tools Appl.*, vol. 76, no. 8, pp. 10881–10892, Apr. 2017.
[7] P. Ekman and W. V. Friesen, "Constants across cultures in the face and emotion," *J. Pers. Soc. Psychol.*, vol. 17, no. 2, pp. 124–129, 1971.
[8] M. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, "Coding facial expressions with Gabor wavelets," in *Proc. IEEE Int. Conf. Autom. Face Gesture Recognit.*, Nara, Japan, Apr. 1998, pp. 200–205.
[9] P. Lucey, J. F. Cohn, T. Kanade, J. M. Saragih, Z. Ambadar, and I. A. Matthews, "The extended Cohn–Kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, San Francisco, CA, USA, Jun. 2010, pp. 94–101.
[10] M. S. Hossain, S. U. Amin, G. Muhammad, and M. Al Sulaiman, "Applying deep learning for epilepsy seizure detection and brain mapping visualization," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 15, no. 1s, p. 1–17, Feb. 2019.
[11] G. Muhammad, M. F. Alhamid, and X. Long, "Computing and processing on the edge: Smart pathology detection for connected healthcare," *IEEE Netw.*, vol. 33, no. 6, pp. 44–49, Nov./Dec. 2019.
[12] Z. Ali, G. Muhammad, and M. F. Alhamid, "An automatic health monitoring system for patients suffering from voice complications in smart cities," *IEEE Access*, vol. 5, pp. 3900–3908, 2017.
[13] H.-W. Ng, V. D. Nguyen, V. Vonikakis, and S. Winkler, "Deep learning for emotion recognition on small datasets using transfer learning," in *Proc. ACM Int. Conf. Multimod. Interact. (ICMI)*, New York, USA, 2015, pp. 443–449.

[14] N. Zeng, H. Zhang, B. Song, W. Liu, Y. Li, and A. M. Dobaie, "Facial expression recognition via learning deep sparse autoencoders," *Neurocomputing*, vol. 273, pp. 643–649, Jan. 2018.

[15] A. Mollahosseini, D. Chan, and M. H. Mahoor, "Going deeper in facial expression recognition using deep neural networks," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Lake Placid, NY, USA, 2016, pp. 1–10.

[16] P. Khorrami, T. L. Paine, and T. S. Huang, "Do deep neural networks learn facial action units when doing expression recognition?" in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2015, pp. 19–27.

[17] E. Barsoum, C. Zhang, C. Canton-Ferrer, and Z. Zhang, "Training deep networks for facial expression recognition with crowd-sourced label distribution," in *Proc. 18th ACM Int. Conf. Multimodal Interact.*, Tokyo, Japan, Nov. 2016, pp. 279–283.

[18] H. Ding, S. K. Zhou, and R. Chellappa, "FaceNet2ExpNet: Regularizing a deep face recognition net for expression recognition," in *Proc. 12th IEEE Int. Conf. Autom. Face Gesture Recognit. (FG)*, Washington, DC, USA, 2017, pp. 118–126.

[19] Y. Guo, D. Tao, J. Yu, H. Xiong, Y. Li, and D. Tao, "Deep neural networks with relativity learning for facial expression recognition," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Seattle, WA, USA, 2016, pp. 1–6.

[20] P. Liu, S. Han, Z. Meng, and Y. Tong, "Facial expression recognition via a boosted deep belief network," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2014, pp. 1805–1812.

[21] M. S. Hossain, and G. Muhammad, "Audio-visual emotion recognition using multi-directional regression and ridgelet transform," *J. Multimodal User Interfaces*, vol. 10, no. 4, pp. 325–333, 2016.

[22] D. K. Jain, P. Shamsolmoali, and P. Sehdev, "Extended deep neural network for facial emotion recognition," *Pattern Recognit. Lett.*, vol. 120, pp. 69–74, Apr. 2019.

[23] R. Zhi, C. Zhou, T. Li, S. Liu, and Y. Jin, "Action unit analysis enhanced facial expression recognition by deep neural network evolution," *Neurocomputing*, vol. 425, pp. 35–148, Feb. 2021.

[24] W. Xiaohua, P. Muzi, P. Lijuan, H. Min, J. Chunhua, and R. Fuji, "Two-level attention with two-stage multi-task learning for facial emotion recognition," *J. Vis. Commun. Image Represent.*, vol. 62, pp. 217–225, Jul. 2019.

[25] P. D. M. Fernandez, F. A. G. Peña, T. I. Ren, and A. Cunha, "FERAtt: Facial expression recognition with attention net," 2019. [Online]. Available: arXiv:1902.03284.

[26] S. Minaee and A. Abdolrashidi, "Deep-emotion: Facial expression recognition using attentional convolutional network," 2019. [Online]. Available: arXiv:1902.01019.

[27] M. S. Hossain and G. Muhammad, "Emotion-aware connected healthcare big data towards 5G," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2399–2406, Aug. 2018.

[28] Y. Zhang, M. Chen, D. Wu, M. S. Hossain, A. Ghoneim, and M. Chen, "Emotion-aware multimedia systems security," *IEEE Trans. Multimedia*, vol. 21, no. 3, pp. 617–624, Mar. 2019.

[29] M. S. Hossain, "Cloud-supported cyber–physical localization framework for patients monitoring," *IEEE Syst. J.*, vol. 11, no. 1, pp. 118–127, Mar. 2017.

[30] A. Ghoneim, G. Muhammad, S. U. Amin, and B. Gupta, "Medical image forgery detection for smart healthcare," *IEEE Commun. Mag.*, vol. 56, no. 4, pp. 33–37, Apr. 2018.

[31] S.U. Amin *et al.*, "Multilevel Weighted Feature Fusion Using Convolutional Neural Networks for EEG Motor Imagery Classification," in *IEEE Access,* vol. 7, no.1. pp. 18940–18950, Feb. 2019.

[32] M. Chen, W. Li, Y. Hao, Y. Qian, and I. Humar, "Edge cognitive computing based smart healthcare system," *Future Gener. Comput. Syst.*, vol. 86, pp. 403–411, Sep. 2018.

[33] G. Muhammad, M. S. Hossain, and N. Kumar, "EEG-based pathology detection for home health monitoring," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 2, pp. 603–610, Feb. 2021.

[34] M. Chen and Y. Hao, "Label-less learning for emotion cognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 7, pp. 2430–2440, Jul. 2020.

[35] M. Chen, Y. Cao, R. Wang, Y. Li, D. Wu, and Z. Liu, "DeepFocus: Deep encoding brainwaves and emotions with multi-scenario behavior analytics for human attention enhancement," *IEEE Netw.*, vol. 33, no. 6, pp. 70–77, Nov./Dec. 2019.

[36] M. S. Hossain, G. Muhammad, and A. Alamri, "Smart healthcare monitoring: A voice pathology detection paradigm for smart cities," *Multimedia Syst.*, vol. 25, no. 5, pp. 565–575, Oct. 2019.

[37] X. Yang, T. Zhang, C. Xu, S. Yan, M. S. Hossain, and A. Ghoneim, "Deep relative attributes," *IEEE Trans. Multimedia*, vol. 18, no. 9, pp. 1832–1842, Sep. 2016.

[38] M. S. Hossain and G. Muhammad, "An audio-visual emotion recognition system using deep learning fusion for a cognitive wireless framework," *IEEE Wireless Commun.*, vol. 26, no. 3, pp. 62–68, Jun. 2019.

[39] M. Alhussein, G. Muhammad, M. S. Hossain, and S. U. Amin, "Cognitive IoT-cloud integration for smart healthcare: Case study for epileptic seizure detection and monitoring," *Mobile Netw. Appl.*, vol. 23, no. 6, pp. 1624–1635, Dec. 2018.

[40] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 86, pp. 2579–2605, 2008.

[41] M. S. Hossain and G. Muhammad, "Cloud-based collaborative media service framework for healthcare," *Int. J. Distrib. Sens. Netw.*, vol. 10, no. 3, Mar. 2014, Art. no. 858712.

[42] L. Hu, M. Qiu, J. Song, M. S. Hossain, and A. Ghoneim, "Software defined healthcare networks," *IEEE Wireless Commun.*, vol. 22, no. 6, pp. 67–75, Dec. 2015.

Ghulam Muhammad (Senior Member, IEEE) received the B.S. degree in computer science and engineering from Bangladesh University of Engineering and Technology, Dhaka, Bangladesh, in 1997, and the M.S. degree from Toyohashi University and Technology, Toyohashi, Japan, in 2003, and the Ph.D. degree in electrical and computer engineering from Toyohashi University and Technology, in 2006.

He is a Professor with the Department of Computer Engineering, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia. He has authored and coauthored more than 250 publications, including IEEE/ACM/Springer/Elsevier journals, and flagship conference papers. He supervised more than 15 Ph.D. and Master Theses. He has two U.S. patents. His research interests include image and speech processing, smart healthcare, and machine learning.

Prof. Muhammad was a recipient of the Japan Society for Promotion and Science Fellowship from the Ministry of Education, Culture, Sports, Science, and Technology, Japan. He received the Best Faculty Award of the Computer Engineering Department at KSU from 2014 to 2015.

M. Shamim Hossain (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of Ottawa, Ottawa, ON, Canada, in 2019.

He is a Professor with the Department of Software Engineering, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia. He is also an Adjunct Professor with the School of Electrical Engineering and Computer Science, University of Ottawa. He has authored and coauthored more than 300 publications, including refereed journals, conference papers, books, and book chapters. His research interests include cloud networking, smart environment (smart city and smart health), AI, deep learning, edge computing, Internet of Things, multimedia for health care, and multimedia big data.

Prof. Hossain is a recipient of a number of awards, including the Best Conference Paper Award, the 2016 *ACM Transactions on Multimedia Computing, Communications, and Applications* Nicolas D. Georganas Best Paper Award, and the 2019 King Saud University Scientific Excellence Award (Research Quality). He is the Chair of the IEEE Special Interest Group on Artificial Intelligence for Health with IEEE ComSoc eHealth Technical Committee. He is on the editorial board of the IEEE TRANSACTIONS ON MULTIMEDIA, IEEE MULTIMEDIA, IEEE NETWORK, IEEE WIRELESS COMMUNICATIONS, IEEE ACCESS, *Journal of Network and Computer Applications* (Elsevier), and *International Journal of Multimedia Tools and Applications* (Springer). He is an IEEE ComSoc Distinguished Lecturer. He is a Senior Member of ACM.