Multimodal Emotion Recognition Using Compressed Graph Neural Networks

Tijana Đurkić[1], Nikola Simić[1](✉), Siniša Suzić[1], Dragana Bajović[1], Zoran Perić[2], and Vlado Delić[1]

[1] Faculty of Technical Sciences, University of Novi Sad, 21000 Novi Sad, Serbia
nikolasimic@uns.ac.rs
[2] Faculty of Electronic Engineering, University of Niš, 18000 Niš, Serbia

Abstract. Since electronic devices have become an integral part of life, there has been a need to bring the communication between a human and a machine closer to being as similar as possible to that between two people. As interpersonal relationships are built on the basis of feelings and empathy, training machines to understand emotions and to provide responses in accordance with the emotional state of the user, i.e. human, has become an interesting area for technology development. To gain a more comprehensive understanding of a person's emotional state, simultaneous utilization of different modalities such as audio, text, and video and their further processing using a graph neural network, recently became popular due to its suitability for tracking a conversation. However, small IoT devices commonly have constrained computational capabilities, memory resources and lower power consumption, and running such a complex multimodal algorithm in real-time may be difficult. In this research, we examine utilization of binarization and 8-bit floating point arithmetic for compressing state-of-the-art GNN-based model COGMEN. We demonstrate that in the case of the multimodal emotion recognition task, such constrained models can provide significant data savings while maintaining relatively high performance, as shown through experiments processing data from the IEMOCAP dataset.

Keywords: Graph Neural Networks · Emotion Recognition · Multimodal Data · Compression

1 Introduction

Interpersonal relationships are significantly influenced by emotional processes, which may play a crucial role in the formation, development, and maintenance of social bonds. Emotions serve to establish trust, build social communities, and gain insight into the causes of certain human reactions [1]. According to Nico Frijda, emotions are the outcomes of a person's engagement with the world and their perception of the environment, which later modifies the reactions and actions they take [2]. By expressing emotions, we obtain information about our interlocutor, learn how someone reacts, and gain the ability to adjust our behavior accordingly.

A. Karpov and V. Delić (Eds.): SPECOM 2024, LNAI 15300, pp. 109–121, 2025.
https://doi.org/10.1007/978-3-031-78014-1_9

Emotions can be conveyed through multiple modalities, one of which is speech communication. The emotional state of a person can be determined based on tone, speech rate, and emphasis on certain words. Prosodic features are objective measures because they encompass everything that makes our speech what it is: frequency, which determines pitch; intensity; spectral envelope, which determines voice color; rhythm; energy; spectral characteristics; and many others [3]. For example, it has been shown that people diagnosed with depression are more prone to using certain words [4]. Therefore, it can be assumed that there is a set of words more commonly associated with specific emotions. Moreover, the meaning of words changes depending on the context, and the flow of conversation influences their choice. It is also possible to clearly assess emotions by observing facial expressions. Some movements are spontaneous, while, for other movements, it has been proven that the same facial muscles contract or relax when experiencing the same emotions in different people [5]. Many of these features are not even consciously recognized, but by observing them, one can determine which emotion is involved.

Sometimes, it is possible to assess a person's emotional state based on a single type of information, but a more complete understanding is obtained by connecting multiple sources, which may require a federated training for achieving better accuracy in the case of sensitive applications [6]. A person can intuitively recognize feelings during communication based on movements, facial expressions, tone, and the words used, even if these emotions are not clearly expressed and named by their name [7].

As electronic devices have become increasingly prevalent in everyday life, people are spending more time using them, which motivated designers to construct machines that can understand instructions in a variety of human moods for a quality user experience. Some of the applications may be critical, as they can involve speaker recognition or other authentication methods, which may be influenced by emotional state of the user [8]. Automatic recognition of the emotional state of users is commonly done from speech, images, video, and text as information sources [9]. By correctly recognizing emotions, machines could adapt their responses. In the case of applications that involve synthetized speech responses, efforts are also being made to ensure that the output voice of a machine has a natural tone, that speech flows smoothly, and includes appropriate pauses in order to further resemble natural human interaction.

Since communication can commonly be performed by processing video, audio, and text, recent trends set a need to effectively process multimodal data to achieve high performance. Recently, GNNs have shown excellent performance and COGMEN model demonstrated better results than traditional models, which are based on convolutional or recurrent neural networks [10]. GNNs are more efficient in representing complex relationships between data, which is significant when it is important to follow the context and more suitable for working with different types of data.

However, achieving their effectiveness comes with the drawback of high computing complexity [11], which leads to higher energy consumption compared to traditional machine learning methods. Processing emotional data locally at the edge reduces the need to send sensitive information to cloud servers, which can enhance user privacy and security [12]. This is particularly important given concerns about data breaches and unauthorized access. As IoT devices at the edge usually have constrained power supplies

and have limited memory, designing quantized models to handle large amounts of data occurred as an important research area [13]. Recently, Ajay et. al proposed a binarized model solution and FPGA-based hardware implementation for real time emotion detection at the edge for passenger anomaly state identification [14]. Another cognitive edge computing architecture involving smartphone and edge server was proposed in [15]. These two studies are focused on emotion detection from facial expressions. However, to the authors' best knowledge there is a research gap in terms of analysis and further hardware implementation of constrained multimodal models, which motivated us to perform initial analysis and examine effectiveness of constrained GNN-based models. Understanding a user's emotional state provides devices with context beyond the explicit content of their messages. This deeper insight can improve device's ability to provide appropriate recommendations or responses, enhancing the overall effectiveness of the interaction.

By reducing the model size through parameter compression, the transfer, storage, and loading of the model are accelerated, enabling the use of applications and services that utilize these models in real time [16]. This further motivated us to examine the effectiveness of applying different quantization techniques to compress state-of-the-art GNN model [10]. Specifically, we examine the use of binarization as a technique that offers a high compression ratio with limited precision, and floating point 8 (FP8) arithmetic, which provides a robust solution capable of delivering quality representation across various input data while significantly increasing computational efficiency compared to full-precision floating point 32 (FP32) arithmetic.

The rest of the paper is organized as follows. In Sect. 2, we provide a brief theoretical background related to the graph neural networks and compression techniques. Section 3 is dedicated to the review of the COGMEN model, which represents a state-of-the-art solution for multimodal emotion recognition. In Sect. 4, we describe quantization techniques that we applied to compress the COGMEN model, and the results are presented in Sect. 5. Finally, advantages and drawbacks of the suggested compression approaches are summed up in Sect. 6.

2 Theoretical Background

2.1 Graph Neural Networks

Graph neural networks are a class of neural network architectures designed to operate with data structured as graphs. Optimal values of the coefficients are determined by perceptron learning and represent the contribution of the inputs. The drawback is that small changes in the inputs can cause large changes in the output (from zero to one and vice versa) even though it is desirable for the output to change in accordance with the magnitude of the changes in the input. To address this, nonlinearity is applied (most commonly, but not necessarily) and the output of the neuron becomes an activation function.

A graph represents connections between a group of entities called nodes. Edges or links connect the nodes and describe their relationships [17]. Graphs are abstract structures of a nonlinear nature, making them very suitable for representing complex relationships between data, while neural networks are used to identify patterns. The

connectivity matrix represents the interconnections between the nodes of a graph and it can be described using Eq. (1). A graph G with n nodes is represented by a connectivity matrix A of dimensions $n \times n$, where a matrix element a_{ij} indicates whether there is an edge going from node i to node j, and a_{ii} represents an edge that goes from and to the same node i. If there is an edge going from node i to node j, then $a_{ij} = 1$, and the other elements are zeros.

$$a_{ij} = \left\{ \begin{array}{l} 1 \text{ if there is an edge from } i \text{ to } j \\ 0 \text{ if no edge exists from } i \text{ to } j \end{array} \right\} \tag{1}$$

In the context of conversation, especially context tracking, a graph with directed edges is used because it is important to follow the influence of utterances on each other. The basic idea of the GNN is to model each node of the graph based on the connections with neighboring nodes, ensuring that complex representations of nodes and the entire graph are learned [17]. This is achieved by iteratively updating the state or features of a node based on information from its neighborhood. Additionally, each edge can have its own attributes. First, information from the surrounding nodes is gathered and combined. This collected data is used to update the state of the node through a function that includes a neural network. The current features of the node are combined with information from its neighbors. As mentioned earlier, the update process is repeated, allowing the properties of distant nodes to influence a node since the information paths traverse the graph multiple times. Training the model involves finding the optimal parameters that allow the network to adjust the weights and achieve the best results for a given problem.

In recent emotion recognition tasks, context is a new feature that is being observed [18], usually by considering text extracted from speech. In this approach, structural connections in the text are observed as features alongside those obtained from audio signals. Two approaches are described. In the first one, attention is given to the speaker's tendency to maintain their emotional state and not succumb to the influence of the environment or other sentences. In the second approach, the focus is on the influence of one speaker on another, and this experience is unique to each person. To better illustrate this complex contextual dependency, the edges of the graph are suitable for their description, while the nodes represent sentences. The advantage of such setup is that, besides the sequential approach where sentences are viewed in isolation from the conversation, it also takes into account the context, considering the influence of each sentence on the conversation.

Intuitively, we form opinions about the interlocutor and their emotional status during a conversation based on what we see, such as facial expressions, and what we hear. If it is a written communication, then we rely on what is written. To take all these aspects into account, a heterogeneous graph [19] is used because its nodes contain data of different modalities. In the work [20], the heterogeneous nodes consist of: sentences providing information about the dialogue history, facial expressions, the speaker's personality type, and the emotional flow during the dialogue. To prove the efficiency of this algorithm, one type of heterogeneous node was removed at a time, showing that the absence of each modality differently affects the overall result but always reduces effectiveness. Another highlighted advantage is that the model learns to automatically recognize emotions and display them accordingly, which, combined with diverse data, would contribute to the development of conversation systems that pay attention to emotions.

2.2 Compression Techniques

The development of machine learning algorithms, especially neural networks, has led to the ability to solve complex challenges. More complex tasks require more layers and parameters. However, large models in terms of the number of parameters are impractical for real-time applications due to limited memory resources and constrained hardware components, e.g., in mobile phones [13]. It has also been shown that these models are over-parameterized, meaning that such a large number of parameters is actually not necessary [21]. For convolutional neural networks and Long Short-Term Memory (LSTM) networks, it has been found that sparse models, on which model pruning has been applied [13], perform better than the initial smaller and denser models, achieving compression by a factor of 10 in terms of non-zero elements with minimal losses in accuracy. Thus, the mentioned study showed that by applying a defined criterion, model pruning is performed, removing a certain number of parameters, making the model sparser and thereby reducing memory and hardware requirements, while the accuracy does not change significantly. Besides sparse models, statistical methods of data compression are widely used. They analyze the input data to determine redundant information and then represents input data with a smaller number of bits. In this paper, two quantization methods are applied and will be explained in the following chapters. We observe binarization technique and floating point-8 arithmetic (FP8), and compare their performance with the one achieved using a full precision 32-bit data.

3 COGMEN Methodology

As a core architecture for development a constrained model we have chosen the COGMEN, which represents a state-of-the-art GNN-based solution for multimodal emotion recognition task [10]. In this section, we provide a short introduction to this existing methodology, before describing steps related to the model compression that we perform and analyze in this paper. The COGMEN supports audio, video, and text modalities that complement each other and provide a more comprehensive inference comparing to the unimodal or other multimodal solutions, available in the literature. What sets COGMEN apart from others on similar or the same topic is that it takes into account the influence of conversation context (global information) and local information, i.e., the interdependence of interlocutors, as well as the dependence on the individual speaker on temporally close sentences. The goal in the experiment was to recognize the emotion expressed in one sentence by the speaker.

To recognize the context (global information) and its impact on each individual sentence (sample), a transformer encoder is used. This approach achieves a better understanding of context and word meaning. Without any positional encoding, the transformer encoder effectively uses the entire context to create distributed representations, describing each sample with multiple features.

Regarding local information, the emotion expressed in one sentence is often influenced by surrounding, neighboring sentences, making it necessary to determine the influence between interlocutors and the influence of the interlocutor on themselves. For these purposes, a graph was developed where each sentence represents a node, and directed edges represent different connections, with the order of sentences being important. It is

crucial to note that one sentence consists of audio, video, and textual representations, making the node multimodal in nature. There is a distinction between the directed connection between sentences spoken by one speaker and the directed connection between sentences coming from multiple speakers.

3.1 Architecture of the Model

The input sentences first encounter the context extractor. The concatenated features of all three modalities (video, audio, and text) are used as input for each sentence of the dialogue, and the context extractor uses a transformer encoder to extract the context. The main component of the transformer encoder is the attention mechanism, which allows the decoder to utilize the most relevant parts of the input sequences by assigning weight coefficients to the encoded input vectors, with the highest coefficients attributed to the most important samples. Three main components are used: queries, keys, and values. The final layer of the encoder is a feed-forward network. In the end, a feature vector is obtained for each sentence.

Based on the features obtained from the encoder, a graph is created that identifies connections between sentences in the dialogue. Each graph node is represented by a sentence in all three modalities, connected by directed edges. Two types of edges can be distinguished: edges connecting sentences spoken by the same speaker, and edges connecting sentences of different speakers. Additionally, future and past relationships of each sentence are tracked, meaning it is known which sentences preceded and which followed each sentence.

The following part of the COGMEN architecture is the Relational Graph Convolutional Network (RGCN) [22]. Its role is to gather transformations specific to the relationships among neighboring nodes, which depend on the type and direction of the edges, all through a normalized sum. In the work of interest, this network model observes the dependency of connected sentences on the speaker and multiple interlocutors. The Graph Neural Network in this work has 52 layers, including layers with weights and layers with biases. It has 8 layers that contain over a million parameters.

To extract rich representations from the node features, a Graph Transformer [23] is used. It adapts the attention mechanism to graph learning by considering nodes connected through edges. At the very end of the entire model, there is an emotion classifier consisting of a single linear layer.

3.2 COGMEN Performance

The performance of the model was examined on two datasets in [10]. IEMOCAP (The Interactive Emotional Dyadic Motion Capture) is a multimodal dataset for emotion recognition where each sentence is labeled with one of 6 emotions: anger, excitement, sadness, happiness, frustration, and neutral feeling [24]. It consists of recorded video dialogues of actors who were asked to act out certain emotions. MOSEI (Multimodal Opinion Sentiment and Emotion Intensity) is another widely used multimodal dataset with 6 emotions: happiness, sadness, disgust, fear, surprise, and anger [25]. As the name suggests, it involves affective emotions expressed in stressful or emotionally challenging

situations. It consists of video and corresponding audio recordings, with each sample labeled with an emotion.

For evaluating the compressed COGMEN model, we have picked IEMOCAP dataset (4-way settings). Such experiment was created in [10], to demonstrate the importance of context. It was created as a sub-dataset by splitting each dialogue into n utterances. It was demonstrated that with the decrease of the number of utterances in the dialogue, there is a certain performance drop. Here, we present detailed results for processing all utterances from each dialogue (the best case) and all three modalities (i.e., audio, video, and text) in Table 1. The results are achieved by training an unconstrained model using the code from the official GitHub repository of COGMEN.

Table 1. Performance of the COGMEN model.

	Precision	Recall	F1-score
Macro-averaged value	82.39%	81.24%	81.66%
Weighted average	82.22%	81.97%	81.96%
Average accuracy	81.97%		

4 Compressed COGMEN Model

Deployment of complex neural network architectures on a resource constrained device, such as mobile platforms, Raspberry Pi and other embedded devices, could heavily depend on the design of constrained models, which commonly involve model quantization. As an extension to the existing work, we perform post-training quantization of the COGMEN model with the goal of saving memory space. We trained the model by using the code from the official COGMEN repositorium. The main objective is to reduce the size of the model by reducing the precision of parameters, primarily weights, which would lead to faster execution, thereby reducing energy consumption as well. We have decided to analyze two popular approaches – binarization as the approach which provides large compression and floating point 8, as an approach which attracts many constrained applications due to the fact that it can provide a 4 times data saving, commonly with only limited drop of other performances. In the next two sections, these approaches are described in detail.

4.1 Binary Quantization

Quantization involves the procedure of mapping input values, which theoretically belong to the infinite set, to an output set with a specific number of values. In this paper, within the quantization process we process the weight coefficients of the network layers. Binary quantization alters the input value using only one bit. Therefore, the complexity of the original data is reduced by converting values into one of two possible values. In this paper, following the methodology presented in [26], we use deterministic function as

the most common approach. It is defined using Eq. (2), where x represents a real input value, while x_b is a binarized sample:

$$x_b = \text{sign}(x) = \begin{Bmatrix} +1 \; x \geq 0 \\ -1 \; x < 0 \end{Bmatrix} \tag{2}$$

One of the challenges of binary quantization, perhaps greater than with other quantization methods, is balancing between data compression and quality preservation. Converting continuous or wide-range values into just two possibilities can lead to the loss of crucial details and information, which in some situations may be unacceptable.

4.2 The 8-bit Floating Point Quantization

Floating point (FP) is the most commonly used method of quantization for representing real numbers in computing. The rules of FP arithmetic are defined by the IEEE 754 standard [27]. According to it, there are three basic binary encodings: with 32 bits, with 64 bits, and with 128 bits. Full precision refers to the 32-bit format, while for the 64-bit and 128-bit formats, it is double and quadruple precision, respectively. The 8-bit FP, also referred sometimes as the minifloat, can be designed following the instructions from the IEEE 754 standard.

As the name suggests, precision is significantly reduced, and such approach is not always suitable for the general-purpose applications, but rather for the special purposes. The drawback of this method is that the range of values representable in FP8 format is limited, which can be problematic when dealing with very small or very large values because using only 8 bits to represent them can lead to loss of accuracy.

Here, an infinite set of input values is represented by a finite set, so complete preservation of arithmetic properties is not possible, and compromises must be made between speed, accuracy, memory saving, implementation, and ease of use. Floating Point numbers consist of three components: the sign bit, exponent, and mantissa (fraction). The sign bit provides information about whether the number is positive or negative, the exponent determines the scale, and the mantissa represents the precision or significant digits. We examine the effects of utilizing the format $(s, e, m) = (1, 5, 2)$, where s, e, and m denote the number of bits allocated for encoding the sign, exponent, and mantissa, respectively. If we consider the input sample x_1, the sign s, and biased exponent E_b are determined by Eqs. (3) and (4), respectively:

$$s = \begin{Bmatrix} 0 \; x_1 \geq 0 \\ 1 \; x_1 < 0 \end{Bmatrix} \tag{3}$$

$$E_b = \left\lfloor \log_2(|x_1|) \right\rfloor. \tag{4}$$

If $E_b > 2^{e-1} - 1$, then the value of E_b is set as $E_b = 2^{e-1} - 1$, and if $E_b < -2^{e-1}$, then $E_b = -2^{e-1}$.

The mantissa M is computed as:

$$M = \text{round}\left(2^m \left(\frac{|x_1|}{2^{E_b}} - 1\right)\right). \tag{5}$$

Also, for $M > 2^m - 1$, it holds that $M = 2^m - 1$ and for $M < 0$, it is $M = 0$. The output value in FP format is obtained using Eq. (6):

$$x = (-1)^s 2^{E_b} \left(1 + \frac{M}{2^m}\right). \tag{6}$$

In the rest of the paper, we perform two experiments as an extension to the existing COGMEN methodology. Firstly, we perform binary quantization and FP8 only to layers with over a million parameters, of which there are 8. This way, we explore the quantization influence on particular layers. In the second experiment, we apply quantization to all layers, and we analyze its' impact on the model's prediction results.

5 Experimental Results and Discussion

In this section, we provide experimental results obtained by applying compression techniques described in Sects. 4.1 and 4.2 to the COGMEN model, and perform a comparison with the performance of the full-precision model. The performance obtained after applying binarization approach from [26] is shown in Table 2. By comparing the results previously shown in Table 1, with those in Table 2 below, a significant drop in model performance can be noticed when binary quantization is applied. It can be noticed that the performance drop is significantly larger in the case of compressing the whole model, comparing to the case of compressing only layers with over million parameters. Previously, the effects of binarization on the performance of GNNs were studied in papers [28, 29]. In the study [28] it was mentioned that there is a slight decrease in accuracy. The impact of quantization on model accuracy is presented in the study [29], where it is stated that the model's accuracy is lower, but insignificantly so, as it is comparable to models on which quantization was not applied. However, a binarization approach, which is used here, was not applied in the study. A more detailed analysis of quantization on the general behavior of the model is given in the study [30]. It is concluded that a non-uniform distribution of data on which quantization is applied affects the performance decline, which may be the cause of such weak results because the histograms show that the value distribution does not match the Laplace distribution around zero, and in that case, binary quantization would have been a suitable method for compression. However, the expected decrease in accuracy was not as significant as that shown in Table 2. Therefore, based on the table, the model's performances are much weaker compared to the original, which was not the case in some earlier experiments that involved GNNs where binarization was applied. Additionally, it's possible that the combination of regular graph neural networks and deterministic binary functions is not predisposed to achieving satisfactory results.

This model architecture is much better suited to the FP8 method for achieving the high-quality performance, as it can be seen in Table 3. The results are even slightly better compared to the results in Table 1 in terms of recall and F1-score, while average accuracy value remains the same.

The reason for such close performance is that with the FP8 method, parameter values are rounded to the nearest quantization levels that are only slightly different from the original values, which is similar behavior as in the model training where parameter values also change in small steps. When accuracy improves slightly, it means that,

Table 2. System performance - binary quantization.

Layers	Over million parameters			All parameters		
	Precision	Recall	F1-score	Precision	Recall	F1-score
Macro-averaged value	49.42%	36.85%	35.25%	38.23%	32.32%	29.03%
Weighted average	48.13%	44.75%	35.25%	39.94%	41.57%	34.64%
Average accuracy	44.75%			41.57%		

Table 3. System performance - FP8.

Layers	Over million parameters			All parameters		
	Precision	Recall	F1-score	Precision	Recall	F1-score
Macro-averaged value	80.90%	81.93%	81.39%	80.99%	82.58%	81.71%
Weighted average	81.86%	81.76%	81.78%	82.15%	81.97%	82.00%
Average accuracy	81.76%			81.97%		

hypothetically, further learning of the model has been applied, as parameter values are adjusted by small amounts corresponding to the learning step, leading to model improvement as seen in [28]. In [30], it was found that using advanced quantization methods like FP8 can lead to accuracy improvements. Also, quantization of biases is suggested to be avoided [30] as they do not require significant storage resources, which is an approach we follow. In the end, it could be noticed that there are not significant performance differences between the cases of compressing the whole model and only several layers with over million parameters. Thus, we can conclude that utilization of the fully compressed model is desirable, due to the larger compression. Based on the above, FP8 has proven to be a suitable method for applying compression to the layers of the graph neural network in this case, as it preserves the results while contributing to the 4 times memory space savings. In Table 4, we present model size of the unconstrained COGMEN model, trained for the 4-way IEMOCAP emotion classification task, and sizes of fully quantized models.

Table 4. Model size.

Precision	Model size (MB)
Full-precision	103.1
FP8	25.8
Binarization	3.22

6 Summary and Conclusions

In this paper, we have examined the influence of the binarization and floating point 8 quantization for compressing GNN-based COGMEN model for multimodal emotion recognition. In the case of the FP8 arithmetic, we have demonstrated that the results are slightly improved and closely approximate to those of the non-quantized model while achieving 4 times memory savings in the case of compressing the whole model. Performance was also tested in the case of applying deterministic binary quantization, which lead to a significant drop in performance, inappropriate for practical implementation. In the future, we will intend to analyze more advanced binarization techniques, such as stochastic binary quantization, as well as a 2-bit scalar quantization in order to examine in details a potential of designing a heavily constrained model.

Acknowledgments. This study was Funded by the European Union (Multilingual and Cross-cultural interactions for context-aware, and bias-controlled dialogue systems for safety-critical applications (ELOQUENCE) project, Grant agreement No. 101135916). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Commission-EU. Neither the European Union nor the granting authority can be held responsible for them.

Also, this research was supported by the Science Fund of the Republic of Serbia, Grant No. 7449, Multimodal multilingual human-machine speech communication, AI-SPEAK.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. De Rivera, J., Grinkis, C.: Emotions as social relationships. Motiv. Emot. **10**, 351–369 (1986)
2. Frijda, N.H.: The Emotions. Cambridge University Press (1986)
3. Delić, V., et al.: Speech technology progress based on new machine learning paradigm. Comput. Intell. Neurosci. **2019**, 1–19 (2019)
4. Yang, C., et al.: Emotion-dependent language featuring depression. J. Behav. Therapy Exp. Psych. **81**, 101883 (2023)
5. Mahlke, S., Minge, M.: Emotions and EMG measures of facial muscles in interactive contexts. Cogn. Emot. **6**, 169–200 (2006)
6. Simić, N., et al.: Enhancing emotion recognition through federated learning: a multimodal approach with convolutional neural networks. Appl. Sci. **14**(4), 1325 (2024)
7. Hebb, D.O.: Emotion in man and animal: an analysis of the intuitive processes of recognition. Psychol. Rev. **53**(2), 88 (1946)
8. Simić, N., et al.: Speaker recognition using constrained convolutional neural networks in emotional speech. Entropy **24**(3), 414 (2022)
9. Cowie, R., et al.: Emotion recognition in human-computer interaction. IEEE Signal Process. Mag. **18**(1), 32–80 (2001)
10. Joshi, A., Bhat, A., Jain, A., Singh, A.V., Modi, A.: COGMEN: COntextualized GNN based multimodal emotion recognitioN. In: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Seattle, USA, pp. 4148–4164 (2022)

11. Liang, F., Qian, C., Yu, W., Griffith, D., Golmie, N.: Survey of graph neural networks and applications. Wirel. Commun. Mob. Comput. **2022**(1), 9261537 (2022)
12. Bajovic, D., et al.: MARVEL: multimodal extreme scale data analytics for smart cities environments. In: proceedings of 2021 International Balkan Conference on Communications and Networking, BalkanCom, Novi Sad, Serbia, pp. 143–147 (2021)
13. Choi, Y., El-Khamy, M., Lee, J.: Universal deep neural network compression. IEEE J. Sel. Top. Sig. Process. **14**(4), 715–726 (2020)
14. Ajay, B.S., Rao, M.: Binary neural network based real time emotion detection on an edge computing device to detect passenger anomaly. In: Proceedings of the 2021 34th International Conference on VLSI Design and 2021 20th International Conference on Embedded Systems (VLSID), Guwahati, India, pp. 175–180 (2021)
15. Muhammad, G., Hossain, M.S.: Emotion recognition for cognitive edge computing using deep learning. IEEE Int. Things J. **8**(23), 16894–16901 (2021)
16. Liu, S., Ha, D.S., Shen, F., Yi, Y.: Efficient neural networks for edge devices. Comput. Electr. Eng. **92**(107121), 1–24 (2021)
17. Wu, L., Cui, P., Pei, J., Zhao, L.: Graph Neural Networks: Foundations, Frontiers, and Applications. Springer (2022)
18. Ghosal, D., Majumder, N., Poria, S., Chhaya, N., Gelbukh, A.: Dialoguegcn: a graph convolutional neural network for emotion recognition in conversation. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP, Association for Computational Linguistics, Hong Kong, China, pp. 154–164 (2019)
19. Zhang, C., Song, D., Huang, C., Swami, A., Chawla, N.V.: Heterogeneous graph neural network. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 793–803 (2019)
20. Liang, Y., Meng, F., Zhang, Y., Chen, Y., Xu, J., Zhou, J.: Infusing multi-source knowledge with heterogeneous graph neural network for emotional conversation generation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 15, pp. 13343–13352 (2021)
21. Neill, J.O.: An overview of neural network compression. arXiv preprint arXiv:2006.03669 (2020)
22. Schlichtkrull, M., Kipf, T.N., Bloem, P., van den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: Gangemi, A., et al. The Semantic Web. ESWC 2018. Lecture Notes in Computer Science(), vol. 10843. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93417-4_38
23. Shi, Y., Huang, Z., Feng, S., Zhong, H., Wang, W., Sun, Y.: Masked label prediction: unified message passing model for semi-supervised classification. In: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21, Montreal, Canada, pp. 1548–1554 (2020)
24. Busso, C., et al.: IEMOCAP: interactive emotional dyadic motion capture database. Lang. Resour. Eval. **42**, 335–359 (2008)
25. Zadeh, A.B., Liang, P.P., Poria, S., Cambria, E., Morency, L.P.: Multimodal language analysis in the wild: Cmu-mosei dataset and interpretable dynamic fusion graph. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, vol. 1, Long Papers, pp. 2236–2246 (2018)
26. Courbariaux, M., Hubara, I., Soudry, D., El-Yaniv, R., Bengio, Y.: Binarized neural networks: training deep neural networks with weights and activations constrained to +1 or –1. arXiv 2016, arXiv:1602.02830v3 (2016)
27. Kahan, W.: IEEE standard 754 for binary floating-point arithmetic. Lect. Notes Status IEEE **754**(94720–1776), 11 (1996)

28. Wang, H., et al.: Binarized graph neural network. World Wide Web **24**, 825–848 (2021)
29. Huang, L., et al.: EPQuant: a Graph Neural Network compression approach based on product quantization. Neurocomputing **503**, 49–61 (2022)
30. Liang, T., Glossner, J., Wang, L., Shi, S., Zhang, X.: Pruning and quantization for deep neural network acceleration: a survey. Neurocomputing **461**, 370–403 (2021)