FVGNN: A Novel GNN to Finger Vein Recognition from Limited Training Data

Jinghui Li[1] ,Peiyu Fang[1]
1. Computer Science, University of Posts and Telecommunications
Beijing, China
echoljh@bupt.edu.cn, fangpeiyu@bupt.edu.cn

Abstract—One cannot make bricks without straw, although deep learning has been widely used, it is a data hungry technique that requires numerous labeled samples. Unfortunately, finger vein dataset has a few images per class which is far from meeting the requirements. To alleviate this problem, considering the powerful ability of graph-based models on relational tasks, we innovatively propose an end-to-end graph neural network(GNN) FVGNN. Images are mapped into embedding node features and then concatenated with labels as inputs. The model learns how to compare inputs, rather than memorize a specific mapping from images to classes. Most of the previous algorithms has a lot of preprocessing and parameter tuning, but in our framework, these are not required. We test our lightweight framework on two well-known datasets, it converges quickly and gets promising results with the accuracy of 99.98%, which outperforms the previous best result.

Keywords—finger vein recognition; graph neural network; embedding network; limited data learning

I. INTRODUCTION

The research area of biometrics is gaining more attention recently, among them finger-vein-based identification is mention worthy due to its efficiency of providing security and accuracy [1], while extrinsic biometric modalities, such as face images, fingerprints and iris, are susceptible to spoof attacks.

The complete process of finger vein recognition system can be generally divided into the following steps: finger vein image acquisition, image preprocessing, feature extraction and feature matching. In data acquisition step, finger vein images are captured using infrared scanner. Captured images are affected by influence of blood pressure of veins, body temperature, and environment circumstances. To get more accurate result, image preprocessing is necessary. Preprocessing step consists of following stages: extracting region of interest (ROI), background removing and image enhancement. Feature extraction is the process of transforming the original finger vein images into more representative numerical feature vectors of the feature space. Feature matching refers to using a certain metric to calculate the distance between two finger vein feature vectors to determine whether they correspond to the same individual.

Extracting the best feature from input data is the most critical step. Recent years have witnessed rapid advances in deep learning [2], achieving impressive results on many large-scale supervised tasks. However, the gradient-based optimization algorithm of neural network requires iterative optimization on a large number of labeled samples to extract powerful representations, so that the model can access excellent generalization performance [3]. Unfortunately, it is often difficult and expensive to acquire large sets of training examples of finger vein images. Most finger vein datasets provide about 10 images of single finger, and the entire dataset contains thousands of images, that is, categories is more numerous than samples in each category. In addition, we need to divide the dataset into training set, validation set and test set, which means fewer images can be used for training. Therefore, applying deep learning in finger vein recognition are still in the trial stage.

As a ubiquitous data structure, graph can capture interactions between individual nodes [4]. As a variant of CNN, graph neural network(GNN) [5], defined with its node attributes and its adjacency matrix, inherits ideas like shared weights and deep hierarchical feature distillation. It learns new representation of a node by aggregating feature vectors of all neighbors, extending the definition of convolution to non-Euclidean spaces. In this paper, the composed GNN outperforms the others on finger vein recognition benchmarks.

The main contributions of this paper are as follows:

1. To our knowledge, this paper is the first to successfully apply the end-to-end graph neural network for finger vein recognition.

2. Propose GNN which successfully solves the problem of the shortage of finger vein images, without complicated data preprocessing and data enhancement.

3. Propose a multi-stage deep neural network which composed of an embedding network, an edge features learning network and the most critical GNN. Compared with the traditional algorithm, this method is more effective.

4. Achieve state-of-the-art performance on finger vein recognition task with fewer parameters and faster model convergence speed.

The rest of the paper is organized as follows: the second section describes some related work. The third part introduces our methodology including notation, model architectures and loss function. The fourth part experimentally proves the effectiveness of our method. Finally, we conclude the paper and point out further work.

II. RELATED WORK

In the finger vein recognition system, feature extraction is one of the most decisive steps. Traditional feature extraction algorithms can be mainly classified into the following: based on

Authorized licensed use limited to: BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE. Downloaded on May 29,2023 at 06:24:23 UTC from IEEE Xplore. Restrictions apply.

finger vein lines, based on finger vein texture features, and based on local invariant features.

The representative methods based on finger vein lines include: repeated line tracking method [6], maximum curvature method [7], average curvature method [8], and Gabor filter method [9].

The method based on finger vein texture utilizes grayscale distribution features within the image. The local binary pattern (LBP) [10] and its variations (for example, LLBP [11], PPBM [12], and PWM [13]) have been used to extract the stable and effective local features, and the Hamming distance is often used as the classification metric. Since texture features statistic local grayscale information, performance is easily affected when there are displacement and rotation in the image.

The method based on minutiae points and local invariant features utilizes topological structures such as endpoints and bifurcation points or local invariant feature descriptors such as SIFT [14], SURF [15], and FREAK [16] for matching and recognition. In theory, they are invariance of translation, rotation, illumination, but due to the poor quality of the finger vein image, it is difficult to extract stable feature points.

In addition to the above algorithms, there are some relatively novel algorithms. For example, based on principal component analysis, manifold space recognition, superpixel, etc. but the recognition process is complicated and the parameters are difficult to adjust.

The tremendous success of deep learning offers a power alternative to hand-crafted features designed by experts. To extract powerful representations, deep neural networks require large quantities samples, as it tends to overfit using a few samples only. To alleviate the gap in some degree, there are already several research directions, including data augmentation [17], transfer learning [18], metric-based method [19-22] and so forth.

Data augmentation is widely used to virtually enlarge the training dataset size and avoid overfitting. By applying a small mutation in the original training data to create new samples, for example, flipping or distorting the input image, adding a small amount of noise, or cropping a patch from a random position. However, models often under-perform when trained on few data. Transfer learning can leverage classifiers already learned, this greatly reduces model parameters and limits over-fitting. But the accuracy will decrease if the target task diverges from the original task. In metric-based method, the inputs are samples and their associated labels, but the outputs are label similarities rather than their associated labels. In essence, it learns a similarity measure, by modeling the distance distribution between samples, the homogeneous samples are close while the heterogeneous samples are far away.

GNN follows the iterative update mechanism, obtaining the new feature vector [23] of the node by aggregating the feature vector of its neighbor node, thereby capturing the structural information in the k neighborhoods of the node. Finally, the representation of the entire graph is obtained by merging.

Applications of GNN can generally be divided into two classes, called graph-focused and node-focused [24]. Graph-focused applications tend to focus on information on an entire graph, including chemical composition studies, image classification, text classification, and more, while the node-focused applications focuse on information of each node in the graph, including object detection, web page classification, and more. This paper belongs to the former, we refer finger vein recognition as a supervised learning task on a graph, where the nodes of graph are images, and edges represent the similarity given by trainable network [25].

III. METHODOLOGY

GNN is asked to predict the label corresponding to the image node in the graph. In this paper, we define an end-to-end graph neural network, each node represents a sample and each edge between pairs of nodes represents the relationship between samples. The model learns not only the node embedding, but also the edge embedding. In detail, first, we use a small CNN network called EmbedNet to obtain the feature vector x of each image i. Then, the edge network EdgeNet models the relationship between all pairs of nodes (x_i, x_j) to obtain the edge embedding, thus building the graph architecture. Fig. 1 illustrates this process. Subsequently, node embedding and edge embedding are input to our graph neural network FVGNN to obtain the updated node embedding. Then continue to update the graph with EdgeNet, update the node embedding with FVGNN, iteratively optimize the model. Finally, model outputs the prediction labels of the samples.

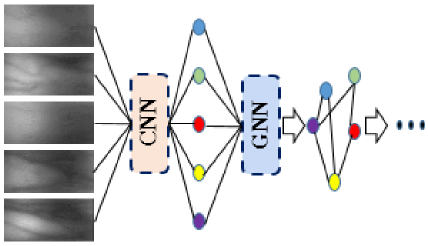

Fig. 1. An illustration of the proposed architecture

A. Basic Notation and Related Definition

We begin by introducing the general setup and our notation. The input-output pairs are $(\Gamma_i; Y_i)$.

The Γ contains a collection of images, both labeled and unlabeled.

$$\Gamma = \{\{(i_1, l_1), .., (i_s, l_s)\}, \{i'\}\}$$
$$Y = (y_1, \dots, y_s) \quad (1)$$

Where $s=q*K$, s is the number of labeled samples, K is the total number of classes and each label appears exactly q times. i' is sample to be classified.

The input feature vectors $(x_1, \dots, x_s)$ is produced by the embedding network EmbedNet from images $(i_1, \dots, i_s)$ in Γ.

To encourage the model to be more adaptable, we use the label smoothing mechanism [26] for the one-hot vector of labels. For a particular example (x_k, y_k), the label distribution $q(k|x) = \delta_{k,y}$ is replaced with the following formula:

$$q'(k|x) = (1-\epsilon)\delta_{k,y} + \frac{\epsilon}{K-1} \quad (2)$$

Where the smoothing parameter ϵ is a small constant.

After label smoothing, the final vector $(l_1, \dots, l_s)$ of labels $(y_1, \dots, y_s)$ is concatenated with the embedding features of the

Authorized licensed use limited to: BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE. Downloaded on May 29,2023 at 06:24:23 UTC from IEEE Xplore. Restrictions apply.

image, by combining domains of label and image, the model can leverage complex dependencies between them.

Let $G = (V; E)$ denotes a fully-connected weighted graph, where node $x \in V$, and edge $e_{i,j}$ means similarity between (x_i, x_j).

B. Model

In this section, we describe the specific model configurations. The model receives two kinds of input:

1) Feature vector x and its label l, given in the form of $N*D$ matrix where N is the number of nodes, D is the dimension of input feature learned by EmbedNet;

2) The description of the graph structure, given in the form of adjacency matrix A learned by EdgeNet;

(a) *EmbedNet Architecture*

EmbedNet computes a M-dimensional dense representation $x_k \in R^M$ of each image $i_k \in R^D$ by an embedding function $f_\omega: R^D \to R^M$ with learnable parameters ω. We tried three structures of VGG [27], Inception [28] and ResNet [29], and selected the best one, the final model structure is shown in Fig. 2. The experiments can be found in the following sections.

The network input is original finger vein image i whose shape is (60, 128), then we use a stack of 4 ResNet blocks, each contains 3x3 conv, batch norm [30] and leaky relu [31], the numbers of 3×3 convolution filters are {64, 96, 128, 256} respectively. The network ends with a fully-connected layer, two dropout layers are used to avoid overfitting. The output x is a 128-dimensional vector.

(b) *EdgeNet Architecture*

The adjacency matrix A is learned by a trained small-size network called EdgeNet. The output is a set of $N*F$ matrixes E, where F is the dimension of output node. The edge features E from the current node hidden representation is computed before every conv layer. The Fig. 3 shows the architecture.

(c) *FVGNN Architecture*

FVGNN iteratively transforms node embedding and edge embedding to the updated node embedding. The finally output y is probability representation of the predicted labels. The Fig. 4 shows the architecture.

C. Optimization

Given a training set $\{(\Gamma_1; Y_1), \ldots, (\Gamma_i; Y_i)\}$, We naturally define a loss function as the following equation:

$$\begin{aligned} \mathrm{L} &= \operatorname{argmin} \frac{1}{I} \sum_{i=1}^{I} l(G(\Gamma_i; \theta), Y_i) + \alpha R(\theta) \\ &= \operatorname{argmin} \frac{1}{I} \sum_{i=1}^{I} \left(-\frac{1}{N} \sum_{n=1}^{N} \sum_{k=1}^{K} y_{n,k} \log h_{n,k} \right) + \alpha R(\theta) \end{aligned} \quad (3)$$

G is our multi-stage model, which outputs a prediction vector from the input matrix Γ_i, where θ is the parameter to be trained. Y is the corresponding ground truth. l is the cross entropy loss. R is a standard regularization including parameter norm penalties and dropout to reduce generalization error. The weight coefficient α is used to balance the loss and regularization.

IV. Experimental Evaluation

A. Dataset

In order to verify the effectiveness of the proposed method and show its advantages over other existing algorithms, we have selected two representative public finger vein datasets.

(a) MMCBNU_6000[32]

The Chonbuk National University finger vein image dataset MMCBNU_6000 contains 6000 ROI finger images with a resolution of 60*128, including the index finger, the middle finger and the ring finger of both hands from 100 individuals, each finger is characterized by 10 images, all of which are grayscale images. In order to facilitate the fair comparison with other algorithms, we will directly select the ROI image provided in the subsequent experiments. The means of images have become zero for normalization.

(b) SDUMLA-FV[33]

The SDUMLA-FV is composed of 3816 finger images from 106 individuals, each of 6 fingers is captured for 6 times per person. This dataset has high intra-class variations in the images and includes significant variations in the quality of images which makes it more challenging to process.

B. Implementation Details

All subnets of our method can be trained synchronously and end-to-end. We set the mini-batch to 64. We used the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, eps $= 10^{-8}$. Learning rate adjustment is crucial for training, we steadily decreased the value from the initial 0.001 which varies over the iteration as follows:

$$lr_{new} = 0.5 \cdot \frac{iter}{10^4} \cdot lr \quad (4)$$

where *iter* represents the current training step. We trained our model on one machine with an NVIDIA 1080Ti GPU.

To further improve the model accuracy, another trick is used. We only apply the regularization to the weights in conv and fc layers, instead of both weights and bias. It is recommended to avoid overfitting [34].

C. Embedded Network Comparison

In this section, we compared three different embedding networks. The input was finger vein image, we employed an additional stack of conv layers that were also learned to convert the image into a feature vector. In order to ensure the credibility, we did it three times independently. The performance on test set is shown in Table I. It shows that the performance of VGG-based and ResNet-based networks is similar, both of which are better than Inception-based networks. Since ResNet-based network has fewer parameters than VGG-based network, ResNet-based network has the best performance in general.

TABLE I. Performance of three different embedding networks on MMCBNU_6000

Authorized licensed use limited to: BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE. Downloaded on May 29,2023 at 06:24:23 UTC from IEEE Xplore. Restrictions apply.

	Accuracy (%)			Iterations	Parameters(M)
	1	*2*	*3*		
Inception-based	99.89	99.90	99.92	190	13.37
VGG-based	99.96	99.97	**99.97**	200	175.12
ResNet-based	**99.97**	**99.98**	99.96	150	60.34

Results demonstrate that this technique greatly improves the expressive capacity and training speed of model, and also that no particular choice of ResNet-based configurations was essential for good performance.

D. Results and Analysis

As the best features are extracted automatically, our method is dataset independent and has a low computation cost and a high computing time efficiency without any special data preprocessing.

The performance measure is based on the percentage of correct classification of the test samples. The results on MMCBNU_6000 are shown as follows. According to the training loss in Fig. 4, it is worth noting that the FVGNN exhibits considerably lower training error and converges fast. As for the test accuracy in Fig. 5, the curve suggests that our model does not experiment overfitting problems, despite the fact that the training set consists of only 10 images per finger. The best result is 99.98%.

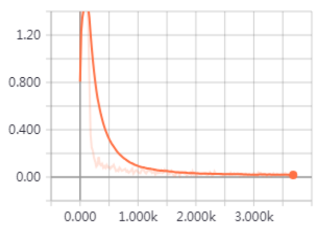

Fig. 4. Training loss

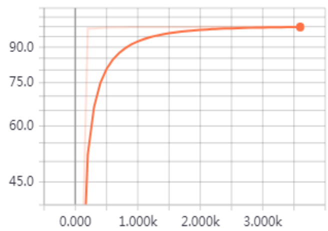

Fig. 5. Test accuracy

To further verify the effectiveness and benefit of the proposed FVGNN, we compared our results with previously reported methods on SDUMLA-FV. The Table II shows the performance. It can be seen that without any complex preprocessing procedures and parameter tuning, after exacting dense representation which has proven to produce better results, our model achieves state-of-the-art performance among all experiments.

TABLE II. COMPARISON RESULTS ON SDUMLA-FV

Reference	ROI Detection	Image Enhancement	Dense representation	Accuracy (%)
Liu Z et al [35]	yes	yes	no	97.80
Wenjie Liu [36]	yes	yes	no	99.53
Gesi Meng [37]	no	yes	no	99.40
This paper	no	no	yes	**99.98**

V. CONCLUSION

In this paper, we presente a strategy for performing finger vein recognition by a novel FVGNN which can efficiently solve the problem of the shortage of data. By way of its corresponding training regime, it is capable of state-of-the-art performance. To best of our knowledge, this is the first ever proposed GNN on finger vein recognition task. This method replaces the similarity metric matching method that is normally applied in common biometric approaches. It is able to quickly distinguish different finger vein images from limited training data. From the results obtained, we can also conclude that data preprocessing and argumentation are not required for the proposed method.

The future research work is the application of our model on more challenging finger vein dataset. In addition, we will conduct research on the combination of different biometric images with the method proposed in this paper.

REFERENCES

[1] J. F. Yang, Y. S. Shi, and J. L. Yang, "Person identification Based on finger-vein features," Computers in Human Behavior, vol. 28, pp. 1565-1570, 2011

[2] Y. LeCun, Y. Bengio, and G. Hinton, Deep learning, Nature, 521(7553):436-444, 2015.

[3] W. J. Scheirer, A. D. R. Rocha, A. Sapkota, and T. E. Boult, "Toward open set recognition," IEEE Transactions on Pattern Analysis & Machine Intelligence, vol. 35, no. 7, pp. 1757-72, 2013.

[4] Hamilton, William L., Ying, Rex, and Leskovec, Jure, Representation learning on graphs: Methods and applications, IEEE Data Eng. Bull., 40:52-74, 2017.

[5] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, Convolutional networks on graphs for learning molecular fingerprints, In Advances in neural information processing systems, pp. 2224-2232, 2015.

[6] Miura N., Nagasaka A., and Miyatake T, Feature extraction of finger-vein patterns based on repeated line tracking and its application to personal identification[J], Machine Vision and Applications, 15(4): 194-203, 2004.

[7] Miura N., Nagasaka A., and Miyatake T, Extraction of Finger-Vein Patterns Using Maximum Curvature Points in Image Profiles[J], IEICE Transactions on Information and Systems, 90(8): 1185-1194, 2007.

[8] Song W., Kim T., Kim H.C., et al. A finger-vein verification system using mean curvature[J], Pattern Recognition Letters, 32(11): 1541-1547, 2011.

[9] Yang J., Yang J., and Shi Y. Finger-vein segmentation based on multi-channel even-symmetric Gabor filters[A], IEEE International Conference on Intelligent Computing and Intelligent Systems[C], pp. 500–503, 2009.

[10] E.C. Lee, H. Jung, and D. Kim, New finger biometric method using near infrared imaging, pp. 2319-2333, 2011.

[11] B.A. Rosdi, C.W. Shing, and S.A. Suandi, Finger vein recognition using local line binary pattern, pp. 11357-11371, 2011

[12] G. Yang, X. Xi and Y. Yin, Finger vein recognition based on a personalized best bit map, pp. 1738-1757, 2012.

[13] Y. Dai, B. Huang, W. Li, and Z. Xu, A method for capturing the finger-vein image using nonuniform intensity infrared light, in: Congress on Image and Signal Processing (CISP), vol 4, pp. 501-505, 2008.

[14] Lowe D.G, Distinctive Image Features from Scale-Invariant Keypoints[J], International Journal of Computer Vision, 60(2): 91-110, 2004.

[15] Bay H., Tuytelaars T., and Van Gool L, SURF: Speeded Up Robust Features[A]. European Conference on Computer Vision[C], pp. 404-417, 2006.

[16] Alahi A., Ortiz R., and Vandergheynst P, FREAK: Fast Retina Keypoint[A]. IEEE Conference on Computer Vision and Pattern Recognition[C], pp. 510-517, 2012.

[17] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In NIPS, 2012.

[18] S. J. Pan and Q. Yang, "A survey on transfer learning," IEEE Transactions on Knowledge & Data Engineering, vol. 22, no. 10, pp. 1345- 1359, 2010.

Authorized licensed use limited to: BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE. Downloaded on May 29,2023 at 06:24:23 UTC from IEEE Xplore. Restrictions apply.

[19] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov, Siamese neural networks for one-shot image recognition, In ICML Deep Learning Workshop, vol 2, 2015.

[20] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al, Matching networks for one shot learning, In Advances in Neural Information Processing Systems, pp. 3630-3638, 2016.

[21] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel, Meta-learning with temporal convolutions, arXiv preprint arXiv:1707.03141, 2017.

[22] Jake Snell, Kevin Swersky, and Richard S Zemel, Prototypical networks for few-shot learning, arXiv preprint arXiv:1703.05175, 2017.

[23] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl, Neural message passing for quantum chemistry, In International Conference on Machine Learning (ICML), pp. 1273-1272, 2017.

[24] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini, The graph neural network model, IEEE Transactions on Neural Networks, 20(1):61-80, 2009.

[25] Victor Garcia, Joan Bruna, Few-Shot Learning with Graph Neural Networks, arXiv preprint arXiv:1711.04043, 2018.

[26] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, Rethinking the inception architecture for computer vision, arXiv:1512.00567, 2015.

[27] Simonyan K., Zisserman A, Very Deep Convolutional Networks for Large-Scale Image Recognition[J], Image Net Challenge, pp. 1-10, 2014.

[28] Szegedy C., Liu W., Jia Y., et al, Going Deeper with Convolutions[A], IEEE Conference on Computer Vision and Pattern Recognition[C], pp. 1665-1672, 2014.

[29] Simonyan K., Zisserman A, Very Deep Convolutional Networks for Large-Scale Image Recognition[J], Image Net Challenge, pp. 1 – 10, 2014.

[30] Sergey Ioffe and Christian Szeged, Batch normalization: Accelerating deep network training by reducing internal covariate shift, ICML, 2015.

[31] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li, Empirical evaluation of rectified activations in convolutional network, arXiv:1505.00853, 2015.

[32] Y. Lu, S. J. Xie, S. Yoon, Z. Wang, and D. S. Park, "An available database for the research of finger vein recognition," Image and Signal Processing (CISP), International Congress, pp. 410-415, 2013.

[33] Y. Yin, L. Liu, X. Sun, SDUMLA-HMT: a multimodal biometric database, Chinese Conference on Biometric Recognition (CCBR), pp. 260 – 268, 2011.

[34] X. Glorot and Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, International conference on artificial intelligence and statistics, pp. 249–256, 2010

[35] Liu Z et al, Finger vein recognition with manifoldlearning, Journal of Network and Computer Applications, 33(3):275–82, 2010.

[36] Wenjie Liu, Weijun Li, Linjun Sun, Liping Zhang, and Peng Chen, "Finger Vein Recognition Based on Deep Learning", IEEE Conference on Industrial Electronics and Applications , pp. 205-210, 2017.

[37] Gesi Meng, Peiyu Fang, and Bao Zhang, "Finger vein recognition based on convolutional neural network", International Conference on Electronic Information Technology and Computer Engineering, 2017.

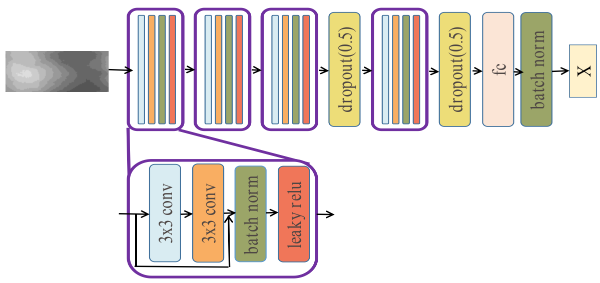

Fig. 2. EmbedNet architecture

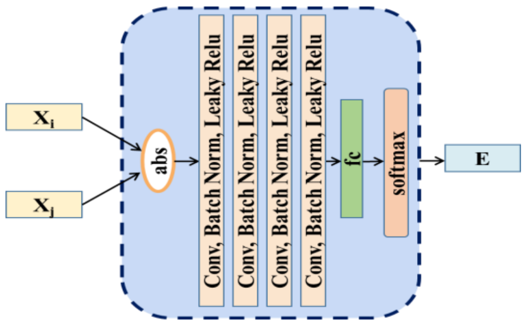

Fig. 3. EdgeNet architecture

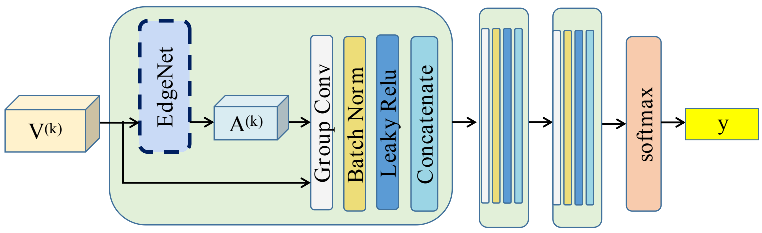

Fig. 4. FVGNN architecture

Authorized licensed use limited to: BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE. Downloaded on May 29,2023 at 06:24:23 UTC from IEEE Xplore. Restrictions apply.