Graph Neural Networks for Interpretable Tactile Sensing

Wen Fan
Dept. of Engineer Mathematics
University of Bristol
Bristol, UK
fanwen2021@gmail.com

Hongbo Bo
Dept. of Computer Science
University of Bristol
Bristol, UK
hongbo.bo@bristol.ac.uk

Yijiong Lin
Dept. of Engineer Mathematics
University of Bristol
Bristol, UK
yijiong.lin@bristol.ac.uk

Yifan Xing
Dept. of Computer Science
University of Bristol
Bristol, UK
yifan.xing@bristol.ac.uk

Weiru Liu
Dept. of Engineer Mathematics
University of Bristol
Bristol, UK
weiru.liu@bristol.ac.uk

Nathan Lepora
Dept. of Engineer Mathematics
University of Bristol
Bristol, UK
n.lepora@bristol.ac.uk

Dandan Zhang*
Dept. of Engineer Mathematics
University of Bristol
Bristol, UK
ye21623@bristol.ac.uk

***Abstract*—Fine-grained tactile perception of objects is significant for robots to explore the unstructured environment. Recent years have seen the success of Convolutional Neural Networks (CNNs)-based methods for tactile perception using high-resolution optical tactile sensors. However, CNNs-based approaches may not be efficient for processing tactile image data and have limited interpretability. To this end, we propose a Graph Neural Network (GNN)-based approach for tactile recognition using a soft biomimetic optical tactile sensor. The obtained tactile images can be transformed into graphs, while GNN can be used to analyse the implicit tactile information among the tactile graphs. The experimental results indicate that with the proposed GNN-based method, the maximum tactile recognition accuracy can reach 99.53%. In addition, Gradient-weighted Class Activation Mapping (Grad-CAM) and Unsigned Grad-CAM (UGrad-CAM) methods are used for visual explanations of the models. Compared to traditional CNNs, we demonstrated that the generated features of the GNN-based model are more intuitive and interpretable.**

***Index Terms*—Tactile Sensor, Object Recognition, Graph Convolutional Network, Explainability.**

I. Introduction

Vision is the major modality for robotic perception, which can obtain global observation of unstructured environments for robots. However, vision-based object recognition may become challenging due to view occlusions or poor lighting conditions. In this case, tactile perception becomes significant since it can provide robots with an alternative exploration mechanism beyond vision. Therefore, we aim to study tactile-based object recognition in this paper.

Among the existing tactile sensors [1], optical tactile sensors have relatively higher spatial resolution [2]. Light conductive plates, reflective membranes, and displaceable markers have been used to construct optical tactile sensors. Among these mechanisms, the marker displacement-based tactile sensors are easy-to-make, since they can be employed in the arbitrary shape of sensor skin and do not require special lighting arrangement. Therefore, we focus on marker displacement-based optical tactile sensors in this paper. TacTip (see Fig. 1(a) and (b)), developed by Bristol Robotics Laboratory (BRL) [3], [4], is such an example that will be used for experimental studies in this paper.

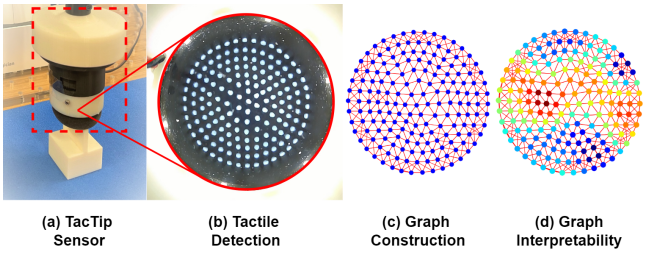


Fig. 1. As seen in (a) and (b), the TacTip's skin will deform when interacting with objects. The embedded camera can capture the pins' movements as tactile information. The tactile graph data can be obtained and analysed based on tactile images, as shown in (c) and (d).

Recent years have seen promising results of combining deep neural network models with tactile perception. For example, Convolutional Neural Networks (CNNs) have been used for texture classification based on a tactile array sensor [5]. CNNs with Long-Short-Term Memory (LSTM) have been integrated into a whole network as CNN-LSTM and applied to a highly-dense optical tactile sensor GelSight [6] for tactile identification of textures. As for TacTip, CNNs have been used for object classification, edge perception, contour following [7], slip detection [8]. However, most of the traditional CNNs trained for object recognition have millions of parameters [9]. For marker displacement-based tactile sensors which have relatively low spatial resolution, CNNs are not efficient for feature extraction. Therefore, more adequate neural network architectures should be explored.

[1]Corresponding author: Dandan Zhang (email: ye21623@bristol.ac.uk).
[2]Wen Fan, Yijiong Lin, Nathan Lepora and Dandan Zhang are affiliated with Bristol Robotics Laboratory.

Recently, Graph Neural Network (GNN) has emerged as an alternative to process irregular data, and has demonstrated better performance compared to CNNs [10]. Considering that the pins (also known as markers) of the TacTip can form a graph [4], while the displacements of pins contain rich information that reveals the contacted objects' shape, we aim to leverage graph-like representations of tactile images for object recognition.

Graph Convolutional Network (GCN) [11], GraphSAGE [12] and Graph Attention Networks (GATs) [13] are representative GNNs. GCN is developed based on applying convolution operation to topological graph, and has been proved to be effective for Physics, Chemistry [14] and social network applications [15], [16]. In this work, we employ GCN-based architecture for tactile object recognition. We aim to compare GCN-based architecture with CNN-based architecture in terms of recognition accuracy and model training speed. Moreover, to ensure that the tactile recognition process is interpretable, Gradient-weighted Class Activation Mapping (Grad-CAM) and Unsigned Grad-CAM (UGrad-CAM) methods are used to enable interpretable tactile sensing [17] [18].

The key **contributions** of our work include:

- Transform the tactile image data obtained by the TacTip sensor into graph representation;
- Develop an optimal GCN-based model for object recognition based on empirical studies;
- Evaluate the intepretability of the GCN-based model for tactile object recognition.

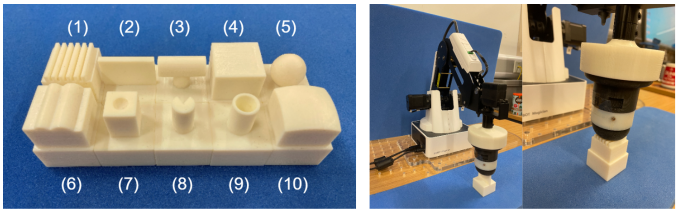
Applying GNN-based methods to vision-based object classification tasks is popular in deep learning. However, to the best of our knowledge, GNN-based tactile-oriented object recognition with biomimetic optical tactile sensors has been scarcely investigated. Intuitive and accurate tactile perception is a prerequisite for manipulation tasks. Based on our results, GNN-based methods can be more efficient for processing tactile images, and have high potentials to benefit tactile robotics research and can ensure interpretability.

We organize the rest of this paper as follows. Firstly, the tactile graph construction and the architecture of the proposed GCN-based frameworks are introduced in Section II. Secondly, the experiment design and results analysis are described in Section III. Finally, conclusions are drawn in Section IV.

II. Methodology

A. Hardware Deployment

Ten different 3D-printed objects are used for data collection, as shown in Fig. 2(a). A low-cost desktop robot arm (Dobot Magician) is used for experiments (see Fig. 2(b)). A TacTip sensor is mounted at the wrist of the robot. The optical tactile sensor TacTip used for data collection comprises a 3D-printed soft rubber-like hemispherical skin, whose papillae pins are distributed uniformly on the inner surface of the skin. The displacements of the papillae pins in the inner surface of the skin can be captured by an RGB camera (ELP 1080p module) to generate tactile data.



(a) Ten Objects to be Classified (b) Data Collection

Fig. 2. (a) Ten 3D-printed objects with different shapes: Grids, Edge, Horizontal Cylinder, Vertical Prism, Sphere, Irregular Shape, Hollow Prism, Irregular Cylinder, Hollow Cylinder, and Curve (from (1) to (10) respectively). (b) A desktop robot arm (Dobot Magician) is used for tactile data collection.

During data collection, a remote controller was used to guide the robot arm with tactile sensor on its wrist to contact the target object's surface, while the tactile image data was recorded simultaneously. The whole dataset for 10 class objects are divided into training set (70%), validation set (20%) and test set (10%) for model training and evaluation.

B. Tactile Graph Construction

As shown in Fig. 1(b), 169 pins of TacTip are distributed uniformly in the shape of concentric circles with increasing radius. Among all the pins, the distances between each pair of adjacent pins are almost identical. The graph representations of tactile images are the inputs for our proposed GCN-based framework. The key features of contact deformation on TacTip sensor will be extracted by the proposed model. All the variables used for graph construction are summarised in Table I.

TABLE I
Parameter Symbol Summary

Parameter	Symbol	Parameter	Symbol
Graph	G	Node position	v
Node	V	Number of edge	m
Edge	E	Number of graph	n
Node feature	X	Source index	s
Adjacency matrix	A	Target index	t

A graph consists of two mandatory components: Nodes (Vertices) and Edges, denoted as $G = (V, E)$. When fed into GNN model, a graph is represented as $G = (X, A)$ where X indicates the node features and A presents an adjacency matrix generated from edges E. The white pins of TacTip can be regarded as graph nodes V, while the positions of the pins in the tactile images are used as node features $X = \left\{v^i = \left(v_x^i, v_y^i\right), i = 1, 2, ...169\right\}$. Let m indicate the total number of edges in one graph. Then undirected edges $E = \{e^j = \left[s^j, t^j\right]^T, s^j \neq t^j, j = 1, 2, ...m\}$ can be built between every pair of possible nodes, where s and t represent the source index and target index respectively. As illustrated in Fig. 3, the graph construction process consists of two steps, i.e. i) node extraction, ii) edge connection.

Every raw tactile image obtained by TacTip sensor is cropped and resized to 280 $\times$ 280, followed by denoising and binarisation (see Fig. 3(a), (b), (c)). Subsequently, the

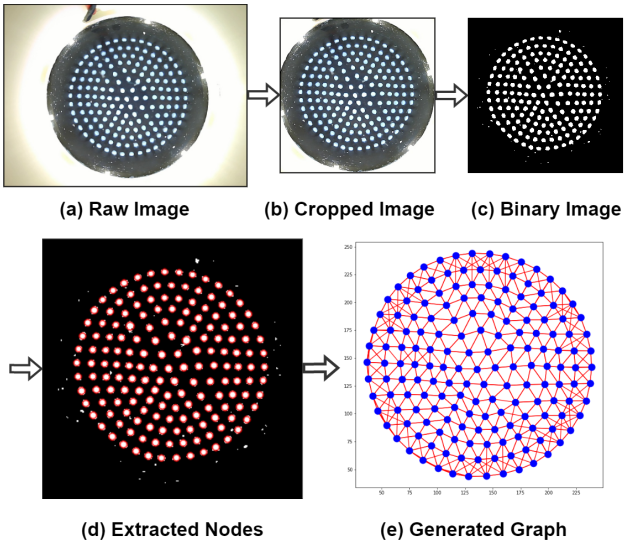


Fig. 3. Tactile graph construction procedure: (a) represents the raw tactile image; (b) shows that the image is cropped and resized; (c) converts the tactile images into binary images; (d) indicates the result obtained after blob detection; (e) presents the final graph obtained after edge connection.

blob extraction algorithm, supplied from the OpenCV library, is used to extract the position of each pin as corresponding node features (see Fig. 3(d)). Finally, the k-Nearest Neighbors (kNN), is used to build edge connections (see Fig. 3(e)).

The quality of generated graph is measured in terms of efficiency, connectivity and robustness. We define that the efficiency is high if the graph construction frequency (the number of graphs generated per second) is higher than 50 Hz. We evaluate graph's connectivity in terms of the neighbors' number linked to every node. High connectivity requires that each node should be connected with at least four adjacent nodes, while redundant connectivity means connection with six adjacent nodes. The robustness is measured based on the difference between the graph connectivity before and after TacTip sensor interacting with objects.

The numbers of nearest neighbors (k value) selected for the kNN classifier have significant impact on the quality of graph construction. According to the characteristics of the TacTip pins' distribution, the outmost nodes should have less neighbors than the inner ones. So we explore an adaptive kNN approach to build edge connections, as shown in Fig. 4. Traditional kNN has a single k value (Fig. 4(a)-(d)), while for our adaptive one, two different parameters k_1 and k_2 are used to cluster nodes in central and non-central areas of the graph respectively (Fig. 4(e)-(f)). The performances of kNN with different k values are summarized in Table II. Adaptive KNN approach can minimize the generation of redundant connections between outmost nodes and remote neighbors. However, the computation time required for the adaptive kNN approach increases significantly, which leads to low efficiency. According to the results, $k = 6$ is selected as the default value for graph construction, which can ensure desired performance in terms of efficiency, robustness and connectivity.

TABLE II
KNN RESULT SUMMARY

Parameter	Edges	Efficiency	Robustness	Connectivity
k = 1	(2, 169)	High	Low	Low
k = 2	(2, 338)	High	Low	Low
k = 3	(2, 507)	High	Low	Medium
k = 4	(2, 676)	High	Medium	High
k = 5	(2, 845)	High	Medium	Redundant
k = 6	**(2, 1014)**	**High**	**High**	**Redundant**
k1, k2 = 6, 5	(2, 971)	Low	High	Redundant
k1, k2 = 6, 4	(2, 928)	Low	Medium	High
k1, k2 = 6, 3	(2, 885)	Low	Medium	High
k1, k2 = 6, 2	(2, 842)	Low	Medium	High
k1, k2 = 6, 1	(2, 799)	Low	Low	Medium

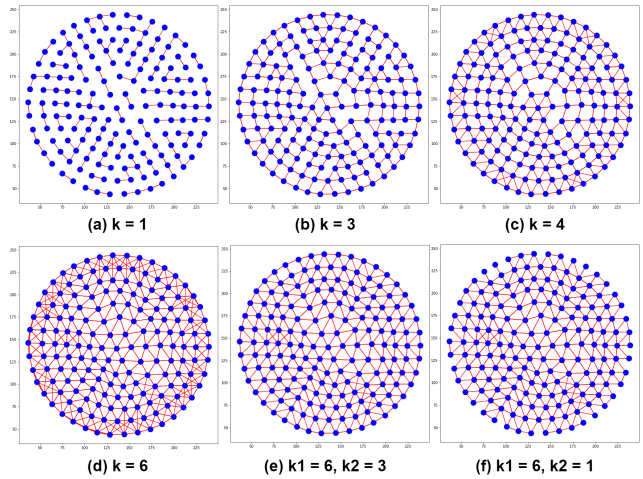


Fig. 4. The graphs generated using kNN with different k value. As shown in (a), (b), (c) and (d), the magnitude of k is proportional to the graph connectivity and integrity. (d), (e) and (f) indicate that adaptive KNN approach can minimize the generation of redundant connections between outmost nodes and remote neighbors.

C. Tactile GNN Framework

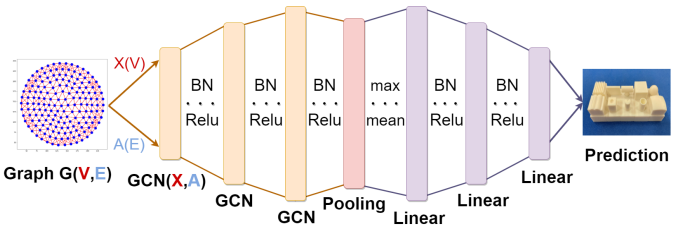


Fig. 5. The framework of Tactile GNN model. The different number of GCN and FC layers, also with two pooling methods, will be tested in the experiment.

The architecture of Tactile GNN model used for object recognition in our paper is shown in Fig. 5. It consists of multiple GCN layers [11], followed by fully-connected layers. A GCN layer can be defined as $H^{'} = GCN(H, \widetilde{A}) = \sigma(\widetilde{D}^{-0.5}\widetilde{A}\widetilde{D}^{-0.5}HW) = \sigma(VHW)$, where $\widetilde{A}$ is the normalized adjacency matrix generated from A, $\widetilde{D}$ is the degree matrix related to $\widetilde{A}$, W is the weight matrix of current GCN layer, and $\sigma(.)$ is the activation function. For example, $ReLU$ is used as the activation function in our work. In the first GCN

layer, $H = X$, where X represents the node features.

We examine the design spaces to choose the optimal structure for the GCN-based model. The results can be found in Section III-A and Table III. Specifically, we experimented with different numbers of GCN and FC layers, and also two pooling methods (scatter-max or scatter-mean).

D. Graph Explainable Methods

Follow the definition of GCN layer $H^{'} = \sigma(VHW)$ and explainable methods for GCN model in work [18], the k'th graph convolutional feature map F at layer l is set as:

$$F_k^l(X, A) = \sigma(VF^{(l-1)}(X, A)W_k^l) \tag{1}$$

The global average pooling feature e of node n from layer L (normally, L is the last GCN layer) should be:

$$e_k = \frac{1}{N}\sum_{n=1}^{N} F_{k,n}^L(X, A) \tag{2}$$

Then the Grad-CAM's weights α for class c is calculated by (3), where the score y of class c is $y^c = \sum_k \omega_k^c e_k$.

$$\alpha_k^{L,c} = \frac{1}{N}\sum_{n=1}^{N} \frac{\partial y^c}{\partial F_{k,n}^L} \tag{3}$$

Finally, the heat-map M which can visualise the positive contribution of node n for graph $G(X, A)$ is generated by:

$$M^c[L, n] = ReLU(\sum_k \alpha_k^{L,c} F_{k,n}^L(X, A)) \tag{4}$$

A new method called UGrad-CAM has been proposed [18], which can show both positive and negative contributions from nodes. We decide to apply both their Grad-CAM and UGrad-CAM explaining tools on our tactile GNN model.

III. Experiments, Results and Discussion

A. Empirical Evaluations

We conduct empirical evaluations to study how the design settings of GCN layers, pooling methods and FC layers influence the performance of Tactile GNN models. We investigate the performance of 4 types of network structures. The first two classes were defined as 'max, original FC' and 'mean, original FC', the last two classes were 'max, standard FC' and 'mean, standard FC'. The 'original FC' meant that each FC layer's channels would vary with the last layer's channels. If using $[(a_i, b_i)](i = 1, 2, ..., I)$ to represent FC structure, then a_i and b_i represented the input and output channel amounts for the i_{th} FC layer, I indicated the total number of FC layers and was set as 3 in this paper. J presented the total number of GCN layers, while c_j represented the output channel numbers for j_{th} GCN layer. For 'original FC', $a_i = c_{J+1-i}$, $b_i = c_{J-i}$, while b_3 was equal to 10. For 'standard FC', $a_1 = c_J$, $a_2 = b_1 = 128$, $a_3 = b_2 = 96$, and $b_3 = 10$. Both pooling methods of scatter-max and scatter-mean were tested to explore their impacts. Adam optimizer (learning rate $\alpha = 10^{-3}$) was used for model training with batch size $\beta = 128$, while early stop mechanism was applied.

The training and evaluation results are summarized in Table III. Based on the results, we notice that the tactile GNN with 7 layers GCN and 3 layers FC has an adequate compromise between computation speed and test accuracy. Increasing the number of GCN layer can enhance the performance, however, the cost of low computation speed is not desirable. The detailed analysis is given below:

1) Depth of GCN: The network prediction accuracy increases when GCN becomes deeper. However, after the depth is greater than 7, the improvement of network prediction accuracy slows down while the cost of training increases dramatically.

Considering the structural features of TacTip and the GNN working principles, the feature aggregation from the center to the outermost nodes (or the opposite direction) would require at least 7 steps (see Fig. 6). This could explain why the predictions improve significantly while the layer number is less than 7. However, network models with 8, 9 and 10 GCN layers were prone to be over-smoothing, which could result in the same representation of most nodes. The time required for training one epoch using 6-layer GCN and 7-layer GCN is 17.4s and 24.9s respectively, which do not have significant difference. Compared to 6-layer GCN, the test accuracy of 7-layer GCN increases 0.4%. The test accuracy for 8-layer GCN is lightly better than the one for 7-layer GCN, while the improvement is less than 0.2%. However, the computation time of the 8-layer GCN is nearly two times longer than that of the 7-layer GCN.

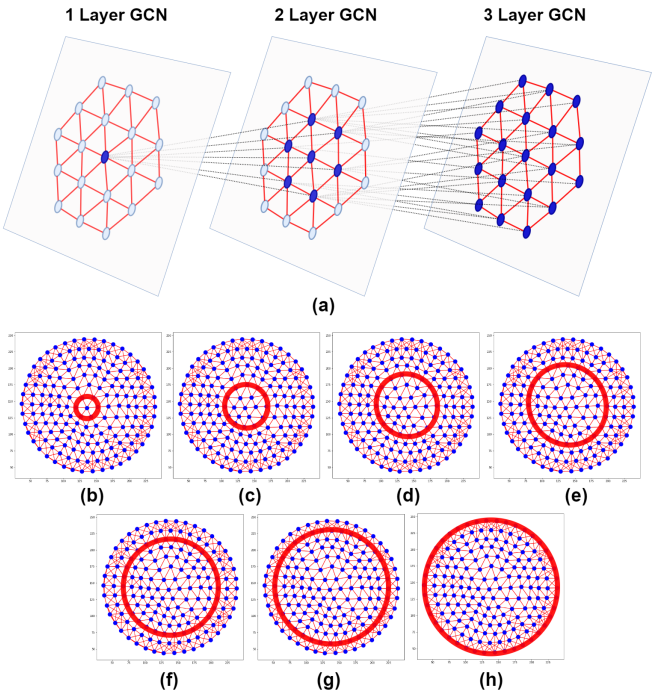


Fig. 6. The aggregation from the center to the outermost circle: (a) presents the aggregation details between GCN layers. Since there are 8 layers of the pins, ideally, at least seven aggregation steps are required to maintain the spatial features, as shown in (b)-(h) respectively.

2) Pooling Methods: Scatter-mean normally performs better than scatter-max. With the same setting of hyperparameters, the training speed of GNNs using scatter-mean are 2% - 5% faster than those using scatter-max. Moreover, GNNs had higher prediction accuracy when using scatter-mean than scatter-max. These differences were evident when Tactile GNN models have 1 or 2 GCN layers. This may be due to the fact that scatter-mean incorporates all node features, which reduces information loss and allows the FC network to learn more useful features.

3) FC layers: The structure of FC layers had significant influence on the model performance. The model with standard FC layers showed better performance than the original FC until the depth of GCN layers increased to 7. When the depth was more than 8, the model with standard FC layers had lightly lower test accuracy compared to the one with original FC layers. However, it had the advantage of higher computation efficiency, since fewer neurons were required for model training.

TABLE III
TACTILE GNN TRAINING AND EVALUATING SUMMARY.

Depth	Class	Train Time (s/epoch)	Test Accuracy
1	max,original FC	5.2	62.31%
1	mean,original FC	5.1	79.90%
1	max,standard FC	5.2	76.60%
1	mean,standard FC	5.0	89.01%
2	max,original FC	6.2	86.87%
2	mean,original FC	6.2	87.97%
2	max,standard FC	6.0	93.87%
2	mean,standard FC	6.0	95.73%
3	max,original FC	7.9	94.52%
3	mean,original FC	7.9	94.62%
3	max,standard FC	7.9	97.29%
3	mean,standard FC	7.8	97.32%
4	max,original FC	10.2	96.64%
4	mean,original FC	10.0	96.84%
4	max,standard FC	10.2	98.41%
4	mean,standard FC	10.0	97.88%
5	max,original FC	13.6	98.16%
5	mean,original FC	13.3	97.95%
5	max,standard FC	13.5	98.67%
5	mean,standard FC	13.2	98.56%
6	max,original FC	18.0	98.48%
6	mean,original FC	17.5	98.63%
6	max,standard FC	17.7	98.70%
6	mean,standard FC	17.4	98.60%
7	**max**	**26.2**	**98.97%**
7	**mean**	**24.9**	**98.99%**
8	max,original FC	42.6	99.03%
8	mean,original FC	39.8	99.13%
8	max,standard FC	42.1	99.04%
8	mean,standard FC	39.8	99.16%
9	max,original FC	75.7	99.35%
9	mean,original FC	73.7	99.35%
9	max,standard FC	74.7	99.13%
9	mean,standard FC	72.9	99.28%
10	max,original FC	151.8	99.49%
10	**mean,original FC**	**150.2**	**99.53%**
10	max,standard FC	151.5	99.10%
10	mean,standard FC	149.9	99.26%

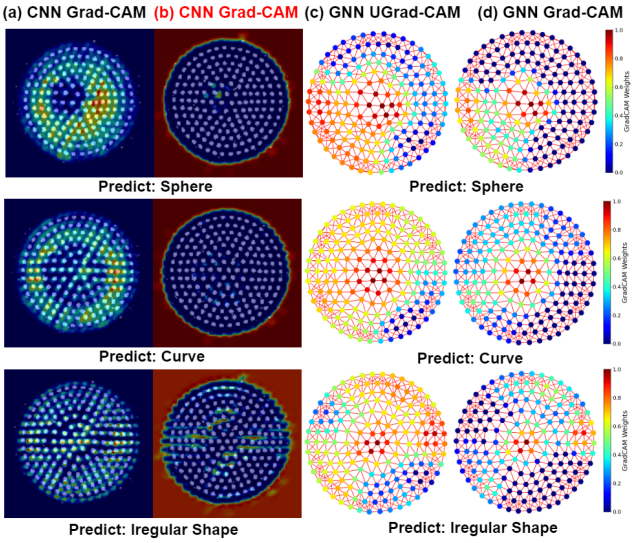


Fig. 7. The visual explanation results using Grad-CAM and UGrad-CAM. (a) shows the examples that CNNs extract useful features from the tactile images; (b) indicates the situation that CNNs shift the main attention to the background, which is not reasonable; (c) UGrad-CAM method is used to visualize the positive and negative impacts from nodes; (d) The positive contributions of each node for GNN to predict is visualized via Grad-CAM.

B. Results Analysis

To ensure that the deep learning-based tactile recognition process is interpretable, Grad-CAM and UGrad-CAM are used to provide visual explanations for GCN models [18].

1) Comparisons for Recognition Accuracy: CNN networks with standard FC layers and different numbers of convolutional layers were constructed for comparisons. The training speed of CNNs was much lower than GNNs with same number of layers, while CNNs' prediction accuracy was slightly higher. A 3-layer CNN provides the best performance with test accuracy of 99.8%. However, the training time for one epoch is 129.2s, which is much longer than 3-layer GCN. If increasing the CNN layers to 7, the training time is 314.9s per epoch while the test accuracy is 99.67%. As for the 7-layer GCN, the test accuracy is 24.9s. This indicates that GCN-based models are computationally efficient.

2) Comparisons for Interpretability: We then compare the interpretability of CNNs and GNNs for tactile perception. The examples of Grad-CAM and UGrad-CAM based analysis are shown in Fig. 7. The red regions refer to the areas where the attention from model is strong. For Grad-CAM based analysis, the blue regions indicate the areas contribute less to the models' decision-making process. As for UGrad-CAM, blue areas represent negative contributions to decision-making.

Fig. 7(a) shows examples that CNNs can successfully extract useful features depending on the deformation. However, they may generate the prediction mostly based on the background information (see Fig. 7(b)), which is not reasonable since the background has been denoised and binarised (see Fig. 4(c)). As for GNNs-based model, attention is paid to the contact regions instead of the background areas. Fig. 7(c)

and (d) show the visual explanations of GCNs-based object recognition process using UGrad-CAM and Grad-CAM respectively. Refer to Fig. 4(d)(e), implicit contact information, such as the location and level of deformation can be visualised.

To quantify the interpretability of both types of models, we randomly selected 100 tactile images for analysis. 55% of tactile images failed to be interpreted by CNNs in a reasonable manner, since the attention of CNNs is located at the background, instead of the deformation. According to the results, GNNs can identify the regions where deformation is caused by interaction with the target object.

IV. Conclusion

In this paper, a GCN-based model (Tactile GNN) was proposed for tactile object recognition. The soft biomimetic optica sensor TacTip was used to record tactile images. The performance of Tactile GNN models with different structures were investigated, while the highest test accuracy for object recognition could reach 99.53%. Grad-CAM and UGrad-CAM were used to evaluate the interpretability of the proposed model. Compared with CNNs, the accuracy for object recognition using GNNs is lower, but the training efficiency is improved significantly. This is due to the fact that Tactile GNNs can extract the key contact information effectively, while CNNs based methods need to process the entire tactile images that contain redundant information. Moreover, Tactile GNN could identify the useful information of deformation, and can be used for interpretable tactile sensing.

We hope the empirical studies and discussions in this paper can provide inspiration for researchers who are interested in GNN-based tactile object recognition with interpretability. We envision that the proposed method can be generalised to other types of tactile sensors and benefits robot learning research where efficient tactile perception is significant.

References

[1] Z. Kappassov, J.-A. Corrales, and V. Perdereau, "Tactile sensing in dexterous robot hands," *Robotics and Autonomous Systems*, vol. 74, pp. 195–220, 2015.

[2] K. Shimonomura, "Tactile image sensors employing camera: A review," *Sensors*, vol. 19, no. 18, p. 3933, 2019.

[3] B. Ward-Cherrier, N. Pestell, L. Cramphorn, B. Winstone, M. E. Giannaccini, J. Rossiter, and N. F. Lepora, "The tactip family: Soft optical tactile sensors with 3d-printed biomimetic morphologies," *Soft robotics*, vol. 5, no. 2, pp. 216–227, 2018.

[4] N. F. Lepora, "Soft biomimetic optical tactile sensing with the tactip: A review," *IEEE Sensors Journal*, vol. 21, no. 19, pp. 21 131–21 143, 2021.

[5] S. S. Baishya and B. Bäuml, "Robust material classification with a tactile skin using deep learning," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 8–15.

[6] R. Li and E. H. Adelson, "Sensing and recognizing surface textures using a gelsight sensor," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1241–1247.

[7] N. F. Lepora, A. Church, C. De Kerckhove, R. Hadsell, and J. Lloyd, "From pixels to percepts: Highly robust edge perception and contour following using deep learning and an optical biomimetic tactile sensor," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2101–2107, 2019.

[8] J. W. James and N. F. Lepora, "Slip detection for grasp stabilization with a multifingered tactile robot hand," *IEEE Transactions on Robotics*, vol. 37, no. 2, pp. 506–519, 2020.

[9] J. M. Gandarias, A. J. Garcia-Cerezo, and J. M. Gomez-de Gabriel, "Cnn-based methods for object recognition with high-resolution tactile sensors," *IEEE Sensors Journal*, vol. 19, no. 16, pp. 6872–6882, 2019.

[10] A. Garcia-Garcia, B. S. Zapata-Impata, S. Orts-Escolano, P. Gil, and J. Garcia-Rodriguez, "Tactilegcn: A graph convolutional network for predicting grasp stability with tactile sensors," in *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019, pp. 1–8.

[11] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *5th International Conference on Learning Representations, ICLR*, 2017.

[12] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, vol. 30, 2017.

[13] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *6th International Conference on Learning Representations, ICLR*, 2018.

[14] K. Do, T. Tran, and S. Venkatesh, "Graph transformation policy network for chemical reaction prediction," in *Proc. of SIGKDD*, 2019, pp. 750–760.

[15] W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, and D. Yin, "Graph neural networks for social recommendation," in *The world wide web conference*, 2019, pp. 417–426.

[16] H. Bo, R. McConville, J. Hong, and W. Liu, "Ego-graph replay based continual learning for misinformation engagement prediction," in *2022 International Joint Conference on Neural Networks (IJCNN)*, 2022.

[17] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.

[18] P. E. Pope, S. Kolouri, M. Rostami, C. E. Martin, and H. Hoffmann, "Explainability methods for graph convolutional neural networks," in *Proc. of IEEE/CVF CVPR.*, 2019, pp. 10 772–10 781.