

# Graph Neural Networks for Interpretable Tactile Sensing

Wen Fan<sup>†§</sup>, Hongbo Bo<sup>‡</sup>, Yijiong Lin<sup>†§</sup>, Yifan Xing<sup>‡</sup>, Weiru Liu<sup>†§</sup>, Nathan Lepora<sup>†§</sup>, Dandan Zhang<sup>\*†§</sup>

<sup>§</sup> Bristol Robotics Laboratory, <sup>†</sup> Department of Engineering Mathematics, <sup>‡</sup> Department of Computer Science  
University of Bristol, Bristol, UK

**Abstract**—Fine-grained tactile understanding of objects is a significant skill for robots to explore and understand the unstructured environment. Convolutional neural networks (CNNs) have been used for tactile perception based on high-resolution optical tactile sensors. However, for tactile sensors which employ cameras to capture pin/marker displacement for contact shape, CNNs using the tactile images contain much redundant information and so are inefficient. In this work, we propose a Graph Neural Network (GNN)-based approach for object identification using a soft biomimetic optical tactile sensor called the TacTip. The obtained tactile images can be transformed into graph format to further analyse the implicit tactile information through GNNs. The experimental results indicate that the proposed method is effective whose object recognition accuracy can reach 99.53%. To evaluate the models’ interpretability, different Grad-CAM methods are used for visual explanations. Compared to traditional CNNs, we demonstrated that the output features of our 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 environment 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 in terms of providing robots with an alternative exploration mechanism beyond vision only. Therefore, we aim to study tactile-based object recognition in this paper.

Among the existing tactile sensors [1], optical tactile sensors have a relatively higher spatial resolution [2]. Optical tactile sensors can convert physical contact (tactile stimuli) into light, then a camera is used to record the physical contact information. Optical tactile sensors have been developed based on a light conductive plate, reflective membrane, and marker displacement. 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 have requirements for special lighting arrangement. We focus on marker displacement based optical tactile sensors for object recognition in this paper. TacTip (see Fig. 1(a) and (b)), developed by Bristol Robotics Laboratory (BRL) [3], [4], is

YL was supported by the China Scholarship Council (CSC)/University of Bristol joint-funded scholarship. NL was supported by a Leadership Award from the Leverhulme Trust on ‘A biomimetic forebrain for robot touch’ (RL-2016-39).

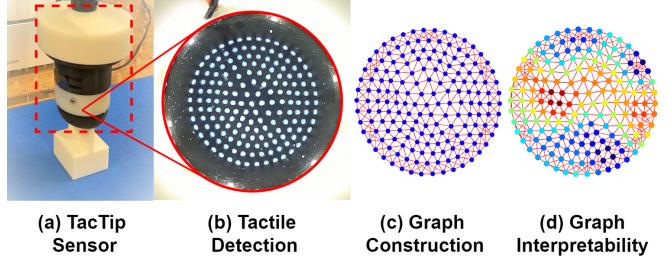


Fig. 1. As seen in (a) and (b), the TacTip’s skin will take place deformation when touching the object. Then, the embedded camera can capture this tactile information through pin tips’ movement. Based on tactile images, we construct the tactile graph data and analyse their interpretability, shown in (c) and (d).

such an example that will be used for experimental studies in our work, while the proposed method can be generalised to different types of tactile sensors.

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], and grasp success prediction for lifting [9]. However, most of the traditional CNNs trained for object recognition can have millions of weights [10]. For marker displacement based tactile sensors which have the relatively low spatial resolution, CNNs are not efficient due to the shortage of features in tactile image data. Therefore, more adequate neural network architectures should be explored.

Recently, Graph Neural Networks (GNNs) have emerged as an alternative to process irregular data and have demonstrated great potential in terms of better performance for processing graph-structured data [11]. Considering that the pins (also known as markers) for the TacTip can form a graph, while the displacements of pins contain rich information that reveals the contacted objects’ shape, so we aim to leverage graph-like representations of tactile images obtained via TacTip for object recognition. Graph Convolutional Network (GCN) [12], GraphSAGE [13] and Graph Attention Networks (GATs) [14] are representative GNNs. GCN is developed based on

applying convolution operation to topological graph, and has been proved to be effective for Physics [15], Chemistry [16] and social network tasks [17]. In this work, we employ GCN-based architecture for graph-based object classification as preliminary studies, while the optimal GNN-based architecture will be explored in the future.

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

- Build an Object Recognition Tactile (ORT) Dataset using TacTip sensor, which will be made publicly available;
- Transform the tactile data obtained by the TacTip sensor into graph representation and develop an optimal GCN-based model for object recognition with empirical studies;
- Evaluate the interpretability of the GCN-based model for tactile-oriented object recognition.

The idea of 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. We call the proposed model as Tactile GNN. This proposed method can benefit robot learning for dexterous manipulation with tactile sensing, since intuitive and accurate tactile perception is a prerequisite for manipulation tasks.

## II. METHODOLOGY

### A. Hardware Deployment

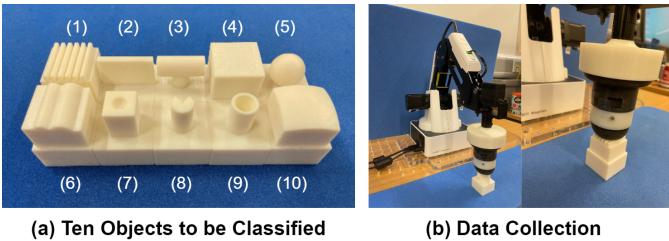


Fig. 2. (a) Ten 3D-printed parts with different shapes: Grids, Edge, Horizontal Cylinder, Vertical Prism, Sphere, Irregular Shape, Hollow Prism, Irregular Cylinder, Hollow Cylinder, and Curve (from (1) to (10) respectively). (b) presents the tactile data collection process with a desktop robot arm.

Ten different 3D-printed objects are used for data collection, as shown in Fig. 2(a). A tactile robot is used for experiments, which comprises a low-cost desktop robot arm called Dobot Magician (see Fig. 2(b)). A TacTip sensor is mounted at the wrist of the robot as an end effector. The morphology of the TacTip used for data collection comprises a 3D-printed soft rubber-like hemispherical skin. 169 hard white tip pins are distributed uniformly on the inner surface in a concentric circular grid fashion. The inner surface of the sensor skin is captured by an RGB camera (ELP 1080p module).

During data collection, the tactile sensor was moved to the position above the 3D-printed object randomly, without aligning the central axis of the tactile sensor. Following that, a remote controller was used to guide the sensor into contact with the object surface, while the tactile image data was recorded simultaneously. The whole dataset for 10 class

objects include 169463 images, which are randomly divided into training set (70%), validation set (20%) and test set (10%).

### 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. In this work, graph representations of tactile images are used as the inputs for our proposed GCN-based framework, in which the key features of contact deformation on TacTip sensor will be extracted. In the following, we will explain how to transform the tactile images into graph-structured data. All the variables we 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, graph should be represented as  $G = (X, A)$  where  $X$  indicates the node features and  $A$  presents adjacency matrix generated from edges  $E$ . The white tip pins locate on TacTip can be regarded as the graph nodes  $V$ , while the pins' positions will be the node features  $X = \{v^i = (v_x^i, v_y^i), i \in [1, 2, \dots, 169]\}$ . Let  $m$  indicate the total number of edges in one graph. Then undirected edges  $E = \{e^j = [s^j, t^j]^T, s^j \neq t^j, j \in [1, 2, \dots, m]\}$  can be built between  $s$  and  $t$  which represent every pair of possible nodes. 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 blob extraction algorithm, supplied from the OpenCV library, is used to extract the positions 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

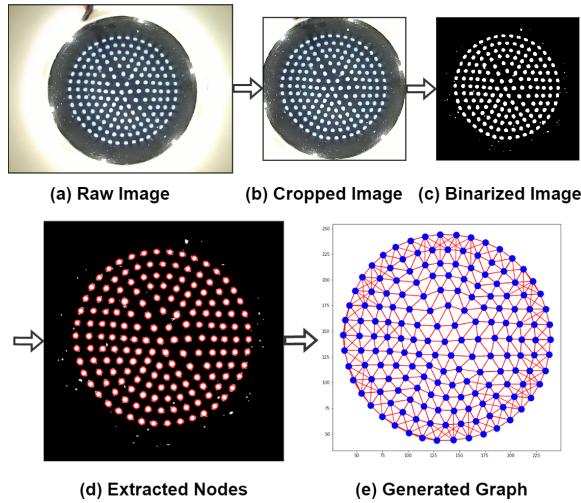


Fig. 3. Tactile graph construction procedure: (a) represents the raw tactile image; (b) shows that the image is cropped and resized; (c) uses masks to deal with overexposed areas around the center, then further apply grayscale and binarisation; (d) indicates the result obtained after blob detection; (e) presents the final graph obtained after edge connection.

construction. According to the characteristics of the tactile image, the outmost nodes should have less neighbors than the inner ones. So we explore an adaptive kNN approach to build edge connection, 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 area of the graph respectively (Fig. 4(e)-(f)). However, the computation time required for the adaptive kNN approach increases significantly, which leads to low efficiency. The performances of kNN with different  $k$  values are summarized in Table II. 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
<b><math>k = 6</math></b>	<b>(2, 1014)</b>	<b>High</b>	<b>High</b>	<b>Redundant</b>
$k_1, k_2 = 6, 5$	(2, 971)	Low	High	Redundant
$k_1, k_2 = 6, 4$	(2, 928)	Low	Medium	High
$k_1, k_2 = 6, 3$	(2, 885)	Low	Medium	High
$k_1, k_2 = 6, 2$	(2, 842)	Low	Medium	High
$k_1, k_2 = 6, 1$	(2, 799)	Low	Low	Medium

### C. Tactile GNN Framework

The architecture of Tactile GNN model used for object recognition in our paper is shown in Fig. 5. It consists of multiple GCN layers [12], followed by fully-connected layers. A GCN layer can be defined as  $H' = GCN(H, \tilde{A}) =$

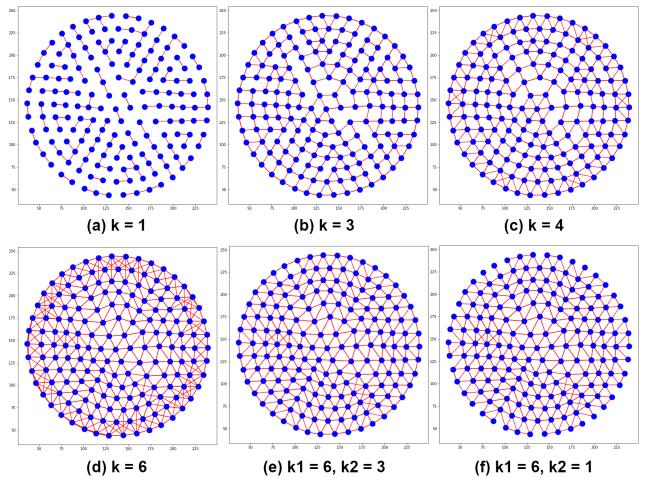


Fig. 4. kNN performance with different  $k$ : As shown in (a), (b), (c) and (d), the magnitude of  $k$  is proportional to the graph connectivity and integrity. Taking (d), (e) and (f) into comparison, adaptive parameters minimize the generation of redundant connections between outmost nodes and remote neighbors.

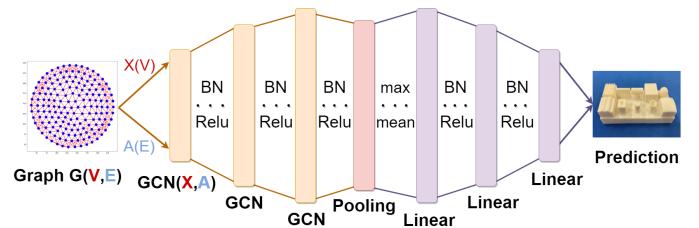


Fig. 5. The framework of Tactile GNN model. The different number of GCN and FC layers, also with two pooling methods, will be tested in the experiment.

$\sigma(\tilde{D}^{-0.5} \tilde{A} \tilde{D}^{-0.5} HW) = \sigma(VHW)$ , where  $\tilde{A}$  is the normalized adjacency matrix from  $A$ ,  $\tilde{D}$  is the degree matrix related to  $\tilde{A}$ ,  $W$  is the weight matrix of current GCN layer, and  $\sigma(\cdot)$  is the activation function, such as  $ReLU$  in our work. In the first GCN layer,  $H = X$  while  $X$  represents the node features.

In this work, we examine the design spaces to choose the optimal structure for GCN-based object recognition based on TacTip, including: 1) various architecture-wise macro designs, 2) various layer-wise micro designs. 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 feasible methods presented in work [18], the  $k$ 'th graph convolutional feature map at layer  $l$  is set as:

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

The output feature of node  $n$  from selected GCN layer  $L$  ( $L$  usually is the last convolutional layer) should be:

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

Based on that, the Grad-CAM's weights for class  $c$  is calculated as follow, where class  $c$ 's score is  $y^c = \sum_k \omega_k^c e_k$ :

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

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

$$H^c[L, n] = \text{ReLU}\left(\sum_k \alpha_k^{L,c} F_{k,n}^L(X, A)\right)$$

Furthermore, they [18] also proposed a new method called unsigned Grad-CAM (UGrad-CAM) 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 trained the model from a random weight initialization using the Adam optimizer (learning rate  $\alpha = 10^{-3}$ ), and applied early stop to model training (patience count = 5), with batch size  $\beta = 128$ .

We conduct empirical evaluations to study how the design settings of GCN layers, pooling methods and FC layers influence the performance of Tactile GNN model. There would be 4 network variants for each GCN depth. We defined first two classes as ‘max, original FC’ and ‘mean, original FC’, then defined last two classes as ‘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. We defined  $[(a_i, b_i)](i = 1, 2, \dots, I)$  to represent FC structure, where  $a_i$  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’,  $a_1 = c_J$ ,  $a_2 = b_1 = 128$ ,  $a_3 = b_2 = 96$ , while  $b_3 = 10$ . Both pooling methods of scatter-max and scatter-mean were tested to explore their impacts on the model performance.

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. From the depth of 1 to 7, the accuracy improvements located about 1% - 10% per layer. After depths upper than 7, this upward trend slowed down considerably, while the cost in time was much higher.

Considering the structural features of TacTip together with the GNN working principles, referred to the Fig. 6, the feature aggregation from the center to the outermost nodes would

require at least 7 steps. This could explain why the predictions improve significantly while the layer number is less than 7. However, deeper networks like 8, 9 and 10 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.

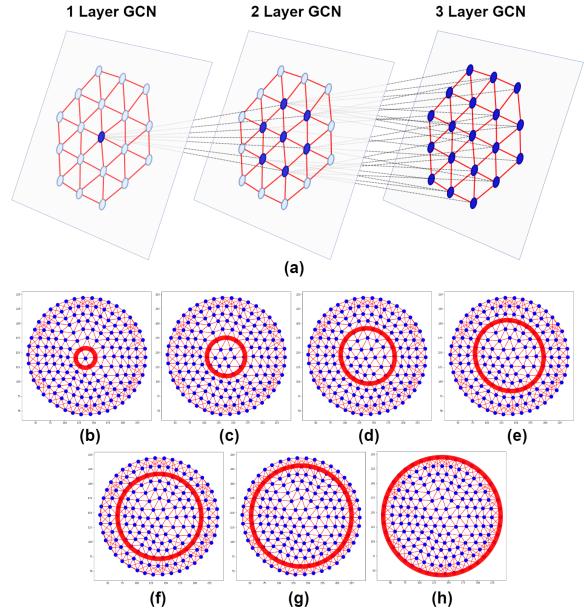


Fig. 6. The aggregation from the center to the outermost circle: (a) presents the aggregation details between GCN layers, since there are 8 layers of the pin tips, ideally it should be at least seven aggregation steps, as shown in (b)-(h) respectively, to maintain the spatial features.

*2) Pooling Methods:* Scatter-mean normally performed better than scatter-max. With same hyperparameters setting, the GNNs using scatter-mean run around 2% - 5% faster during training than those using scatter-max both on original FC and standard FC networks. Also, GNNs enjoyed higher prediction accuracy when using scatter-mean than scatter-max. These improvements were most evident when GCN layers were 1 or 2. It is possible that scatter-max selects only part of the node information from GCN outputs, while 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 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 small computer memory storage, since significantly 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	<b>26.2</b>	<b>98.97%</b>
7	mean	<b>24.9</b>	<b>98.99%</b>
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%
<b>10</b>	<b>mean,original FC</b>	<b>150.2</b>	<b>99.53%</b>
10	max,standard FC	151.5	99.10%
10	mean,standard FC	149.9	99.26%

### B. Explainable Analysis

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

1) *Comparison between CNN and GNN*: CNN networks with standard FC layers and different numbers of convolutional layers were constructed for comparisons. The training speeds of CNNs were much slower than GNNs with same number of layers, while the prediction accuracy of CNNs was 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 increase the CNN layers to 7, the training time is 314.9s per epoch while the test accuracy is 99.67% test accuracy. As for the 7-layer GCN, the test accuracy is 24.9s. This indicates that CGNs-based models are computationally efficient. We then used Grad-CAM methods to compare the interpretability of them.

The examples of using Grad-CAM and UGrad-CAM for analysis of the decision process of CNNs and GCNs are shown in Fig. 7. The red regions refer to the part where the attention from model is strong, while the blue regions indicate the areas

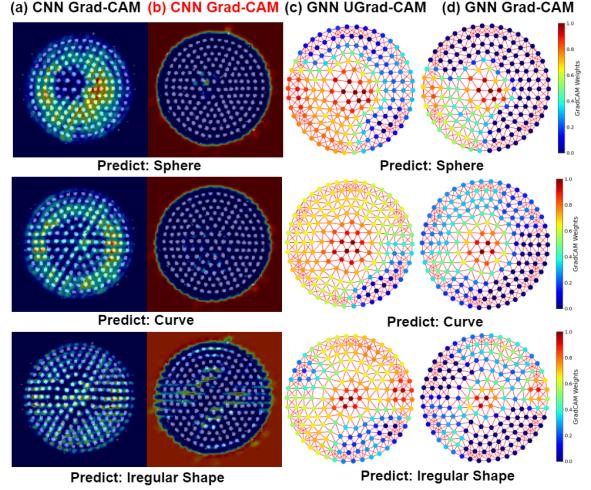


Fig. 7. The visual explanation results using Grad-CAM and UGrad-CAM. (a) shows the examples that CNNs extract useful features from the tactile images; (b) indicates the situation that CNNs shift the main attention to the background, which is not reasonable; (c) UGrad-CAM method is used to visualize the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

that do not contribute to the model's decision-making process (Grad-CAM), or even provide negative contributions (UGrad-CAM). Fig. 7(a) shows that CNNs can successfully extract useful features for recognition based on the deformation.

However, CNNs 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 region instead of the background image. Since the tactile graph constructed by pin nodes and edges (see Fig. 4(d)(e)), both of which totally represent the implicit contact information, such as the precise location of the deformation and the extent to which it occurred. We conduct statistics analysis with 100 randomly selected tactile image. 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 tip pins displacement. Fig. 7(c) and (d) show the visual explanation of GCNs-based object recognition process using UGrad-CAM and Grad-CAM respectively. According to the results, GCN layers can identify the regions of the tactile images where deformation is caused by interaction with the target object.

2) *Comparisons among Models with Different Depth*: We conduct additional experiments to explore how models with different number of GCNs layer learn to capture key features of tactile graphs that can recognise the target objects. As presented in Fig. 8, the model with 1-layer GCN fails to perceive accurately, but there is a correlation between the direction of the nodes' weight distribution and the shape of the actual deformation; the model with 3-layer GCN can locate the general areas of features preliminary; the model with 5-layer GCN can extract almost all the useful features; both models with 7 or 9 GCN layers can locate the key features accurately,

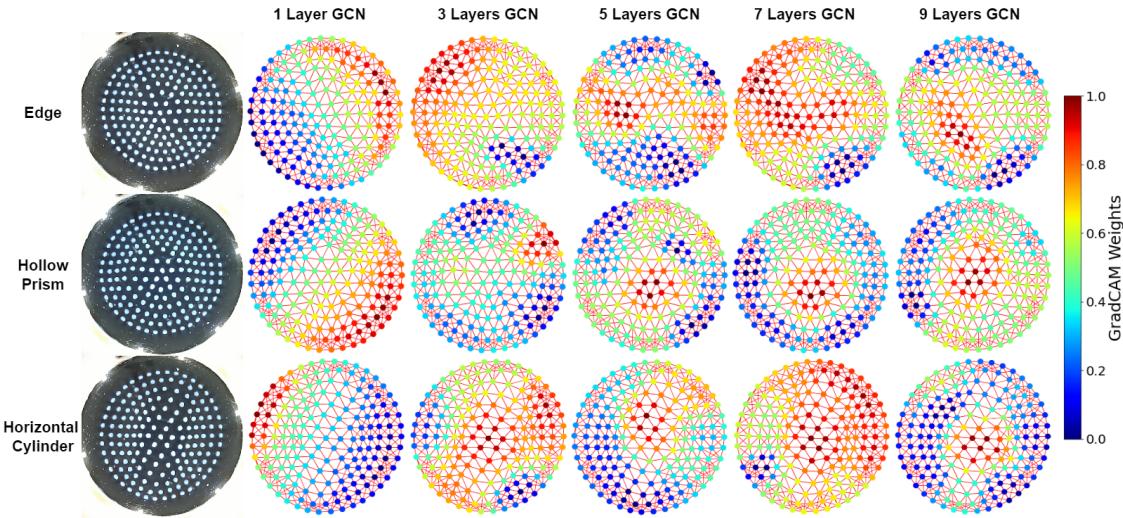


Fig. 8. GNN UGrad-CAM results for GCN layers from 1 to 9. With GCN layers getting deeper, the features learned by the GNN were more accurate. Their activated regions were gradually focusing on the contact deformation areas. When GCN layers added upper than 7 in continue, the variations of activated regions which included key features were no longer evident, but would change in area size and activation level.

while model with 7-layer GCNs focus on a wider range of features compared to that with 9-layer GCN.

#### IV. CONCLUSION

In this work, a tactile-based GNN method was proposed to realise the object classification task. The kNN method was used to achieve tactile graph construction from the output images of the TactTip sensor. And we explored the impact of different parameters in kNN on our generated graph data. Also, the performance of Tactil GNN models with variant layer depths and structure was investigated whose best test accuracy could reach 99.53%, and further analysed by explainable methods, such as Grad-CAM and UGrad-CAM. Compared with CNNs, GNNs could be more efficient in training time for small loss of accuracy. Also GNNs had the benefit that they extracted information primarily from the contact region, whereas CNNs used the entire tactile image. Depending on the GCN layer's output features, we examined variant trends in activation areas for different GCN layers through heat map visualisation. We hope our explorations and discussions can provide inspiration and experience for those researchers from other groups, when using our approach with their different types of tactile sensors.

#### 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. Giancaccini, 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*, 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. W. James, A. Church, L. Cramphorn, and N. F. Lepora, “Tactile model o: Fabrication and testing of a 3d-printed, three-fingered tactile robot hand,” *Soft Robotics*, vol. 8, no. 5, pp. 594–610, 2021.
- [10] 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.
- [11] A. Garcia-Garcia, B. S. Zapata-Impata, S. Orts-Escalano, P. Gil, and J. Garcia-Rodriguez, “Tactilegen: 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.
- [12] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” *arXiv preprint arXiv:1609.02907*, 2016.
- [13] W. Hamilton, Z. Ying, and J. Leskovec, “Inductive representation learning on large graphs,” *Advances in neural information processing systems*, vol. 30, 2017.
- [14] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, “Graph attention networks,” *arXiv preprint arXiv:1710.10903*, 2017.
- [15] A. Sanchez-Gonzalez, N. Heess, J. T. Springenberg, J. Merel, M. Riedmiller, R. Hadsell, and P. Battaglia, “Graph networks as learnable physics engines for inference and control,” in *International Conference on Machine Learning*. PMLR, 2018, pp. 4470–4479.
- [16] K. Do, T. Tran, and S. Venkatesh, “Graph transformation policy network for chemical reaction prediction,” in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 750–760.
- [17] 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.
- [18] P. E. Pope, S. Kolouri, M. Rostami, C. E. Martin, and H. Hoffmann, “Explainability methods for graph convolutional neural networks,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10772–10781.