Multi-purpose Tactile Perception Based on Deep Learning in a New Tendon-driven Optical Tactile Sensor

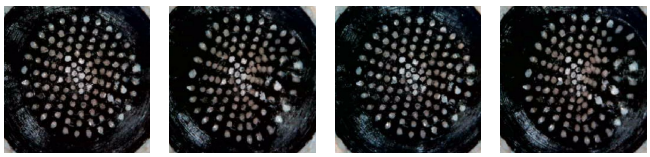
Zhou Zhao[1] and Zhenyu Lu*[2]

***Abstract*— In this paper, we create a new tendon-connected multi-functional optical tactile sensor, MechTac, for object perception in field of view (TacTip) and location of touching points in the blind area of vision (TacSide). In a multi-point touch task, the information of the TacSide and the TacTip are overlapped to commonly affect the distribution of papillae pins on the TacTip. Since the effects of TacSide are much less obvious to those affected on the TacTip, a perceiving out-of-view neural network (O^2VNet) is created to separate the mixed information with unequal affection. To reduce the dependence of the O^2VNet on the grayscale information of the image, we create one new binarized convolutional (BConv) layer in front of the backbone of the O^2VNet. The O^2VNet can not only achieve real-time temporal sequence prediction (34 ms per image), but also attain the average classification accuracy of 99.06%. The experimental results show that the O^2VNet can hold high classification accuracy even facing the image contrast changes.**

I. INTRODUCTION

Tactile sensors can measure tiny deformation of the surface and the pressure created by physical interaction with the object and the environment, and are widely used for robotic manipulation. Due to the low producing cost (only cameras and 3D printing components) of acquiring multiple kinds of information of force distribution, object location, pose, size, and shape, etc., vision-based tactile sensor draw a great of attentions in recent years. The representative optical tactile sensors include BRL TacTip series [1] and Tacto [2]. These tactile sensors usually have a soft skin, feature markers (e.g., papillae pins), and a camera system inside the sensor to capture the image changes when the soft skin interacts with unknown objects. However, the effective sensing ranges are limited to the camera's field of view, so it is not easy to use optical tactile sensors for detection in a large range and make flexible robot skill.

In this work, we will develop a new vision-based tactile sensor: MechTac (see Fig. 2), which combines mechanical transmission and optical detection to increase the tactile regions. The sensor is made of a TacTip and four TacSide skins (see Fig. 3). Under the TacSide skin, the tendons are weaved into nets (see Fig. 2), and add connects into the TacTip area with papillae pins. Then, when you press on different regions on the skin (e.g. in Fig. 2), the associated tendons are tightened to drive the connecting nodes, so that the functional pins are moved under these tendon-driven affections. The MechTac will receive both information from the TacTip and the TacSide.



(a) (b) (c) (d)

Fig. 1. The distribution of papillae pins on the TacTip. (a) the original image; (b) under the direct effect of interaction on the TacTip; (c) under the indirect effect of the TacSide; (d) under the joint effect of the interaction on both the TacTip and the TacSide.

The effect of the TacSide through tendons is much weaker than the one direct interaction with the TacTip. In Fig. 1, compared with the original image (Fig. 1(a)), we can see a clearer deformation under the effect of interaction on the TacTip (Fig. 1(b)) than the indirect effect of the TacSide (Fig. 1(c)). Under the joint effect of the interaction on both the TacTip and the TacSide (Fig. 1(d)), the effect of the TacSide can be easily masked by that of the TacTip. Therefore, based-on monocular images, the core challenge for signal processing is to realize the classification of the independent signal from the hybrid information. And with changes in the distribution of papillae pins on the TacTip, the MechTac captures uneven lighting images as in Fig. 1(b) and Fig. 1(c), which results in unstable image contrast.

Deep learning methods have applied to the object recognition [3], effective estimation of contact force [4], slip and rotation detection [5]. These deep learning methods are mainly data-driven, and study the transformation relationship between the input image and the corresponding label. Hence, for vision-based tactile sensors using deep learning, the pivotal point is to map directly from visual sensor data to task representations. However, most deep learning models are based on monocular images, and the images often contain only a single information to ensure good task performance. In [6], using the monocular camera captured the deformation images in contact region, and input the modified LSTM network to detect contact slip. When other information is needed, it is acquired with other specific cameras. For example, Padmanabha et al. [7] used multiply cameras to capture the images of different view, and then used two different modified ResNet framework for the state estimation and connector insertion tasks, respectively. Trueeb et al. [8] captured images of the different particle patterns through four cameras, and then mapped to the 3D contact force

[1]Zhou ZHAO is with EPITA Research and Development Laboratory (LRDE), Le Kremlin-Bicêtre, France zz@lrde.epita.fr

[2]Zhenyu LU is with Bristol Robotics Laboratory, UWE Bristol, UK zhenyu.lu@uwe.ac.uk

*Corresponding author.

Fig. 2. MechTac. For solving the perceiving tactile information out of field of view (blind area of vision), we use tendons to weave on a net connect the TacTip and the pressing regions. when you press on different regions on the skin the associated tendons are tightened to modify the distributions of the pins on the TacTip to generate displacement of feature pins.

distribution by one deep learning framework. Choi et al. [9] used the depth sensor to obtain partial point clouds of the objects, which included invariance of photometric variations and geometric information, and finally a 3D CNN method is used to grasp unknown objects. As far as we know, given visual information's important role in tactile sensors, directly using deep learning methods to classify different tasks from hybrid information of monocular images (like the hybrid information appearing on the MechTac) has not yet been explored. Moreover, vision-based tactile sensors always face variable environments such as the light intensity resulting in the image contrast changes, but the above-mentioned deep learning methods do not take the image contrast changes into account when they design network frameworks, so these methods cannot maintain good classification performance facing the image contrast changes.

Therefore, to achieve multi-task classification from hybrid information of monocular images and handle the image contrast changes, we make three following contributions: 1) Creating the MechTac sensor and building the training dataset. We collect a total of 10,051 images while the MechTac performs a contact motion. 2) We propose one new perceiving out-of-view neural network, O^{2}VNet, which consists of one proposed binarized convolutional (BConv) layer, one modified DenseNet121 architecture [10], and some fully connected (FC) and dropout layers. The O^{2}VNet can achieve localization of touching regions on the TacSide and objects perception on the TacTip from only one image. Compared with the 3D network proposed in [11], our proposed 2D network also uses the temporal information by stacking three consecutive images as input, instead of high memory consumption and time-consuming 3D network, and accomplishes the multi-tasking separation from the monocular images. 3) Since the MechTac is often subjected to changes in external light intensity when capturing images, the contrast of the image changes, so based on Local Binary Pattern (LBP) [12], we propose a binarized convolutional (BConv) layer that transforms the grayscale image into an image of integer labels, and simply stacks in front of the backbone of the O^{2}VNet for end-to-end training.

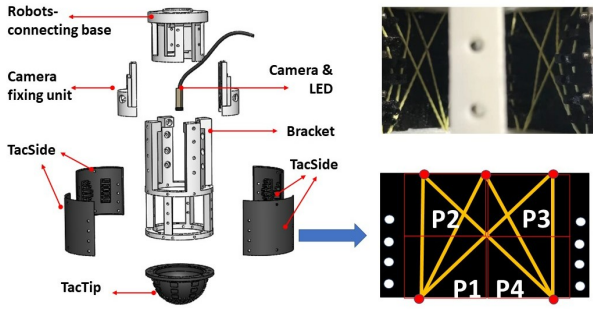


Fig. 3. Mechanical Structure of the MechTac

II. Preliminary Work

A. Mechanical Structure of Tactile Sensor

The mechanical integration in the MechTac is shown in Fig. 3. We choose the hemispherical sensor head for the TacTip to connect the tendons and the connecting nodes are distributed symmetrically, and each TacSide is a thin skin of a quarter cylinder with fixed nodes to attach the tendons to create the mesh structure. An endoscopy camera is clamped by the fixing unit and attached as a whole to the base, which is connected to the robot end-effector. The bracket has three functions: It fixes the camera to the base unit, adjusts the distance between the camera and the TacTip, and is used to mount the TacTip and TacSides. The MechTac has four TacSides (F1 to F4), and in each TacSide the connection topology under the skin in Fig. 3. In each TacSide, there are six tendons that weave the mesh (orange lines). We divide a TacSide into four areas, labelled P1 to P4. Each area covers three lines, which means that each pressing area leads to a co-reaction of three tendons. We will implement three classification tasks using the MechTac (see Fig. 4), including object classification (T1 to T3) and localization of touching region (rough/precise localization, task F and P).

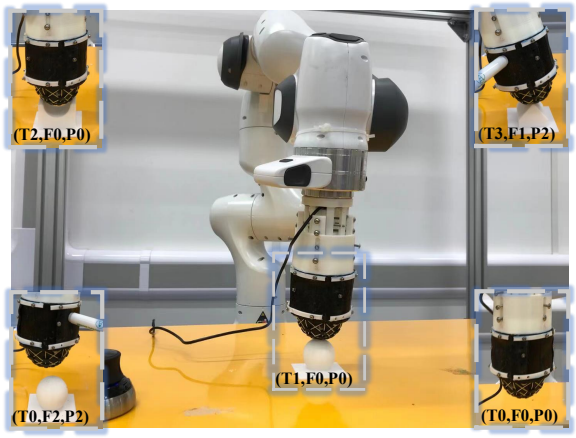


Fig. 4. Tactile data collection. Using Franka robot equipped with MechTac sensor to collect images with labels in Eq. 1

B. Dataset Description

We need to collect tactile image data through the MechTac installed at the robot end-effector, called **TAC** dataset (see Fig. 4). To reduce the effect of class imbalance, we follow certain rules (Eq. 1) to collect data while the MechTac performs a contact motion, for example, when $i = j = k = 0$, we collect one image and label its category (T0, F0, P0). Class T0 denotes that the TacTip dose not touch any object. Class F0 and P0 denote that the TacSide is not subject to touch. The rules include the four cases shown in Fig. 1, which provides O^{2}VNet with the feature distinction of learning the four cases, and is beneficial for O^{2}VNet to locate the touch region under the simultaneous effect of the TacTip and the TacSide. According to the rules, a total of 10,051 images were collected. Each image keeps the same image size (875×656 pixels). Later, we perform some preprocessing on these images: 1) we crop initial images size 875×656 pixels into images size 520×520 pixels; 2) we resize the images size into 224×224 pixels. Finally, we split these images into a training set (8,041), a validation set (1,005), and a test set (1,005).

$$Data(Ti, Fj, Pk) = \begin{cases} 150, & 0 \leqslant i \leqslant 3, 0 \leqslant j, k \leqslant 4 \\ 1, & i = j = k = 0 \end{cases} \quad (1)$$

III. METHODOLOGY

A. Overview of Network Architecture

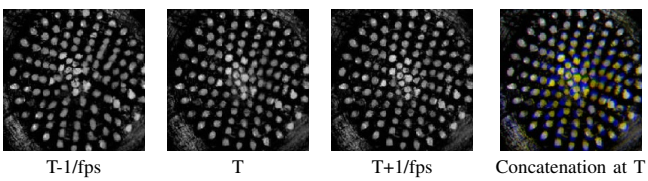


Fig. 5. Illustration of temporal information. fps denotes frame per second.

The overview of the O^{2}VNet is shown in Fig. 7. Touch occurrence is a spatio-temporal event on the skin of MechTac, and it is inevitable that the constructed network uses the spatio-temporal information. 3D networks can make good use of spatio-temporal information by exploiting 3D features while with time-consuming and high memory consumption [13]. Therefore, we prefer 2D networks. Using the temporal information in 2D networks is already common [14], so we take three consecutive images (T-1/fps, T, T+1/fps) as input of the O^{2}VNet and the corresponding label at time T as output, as shown in Fig. 5, motion changes are highlighted by concatenating three consecutive images (see the yellow part of the concatenated image).

The concatenated images contain obvious edge and contour movement information. Convolutional Neural Networks (CNNs) can reliably perceive edge and contour following [15] and learn better performing deep models via transfer learning from ImageNet [16]. Hence, we choose the original DenseNet121 [10] network architecture as the backbone, pretrained on ImageNet. We then discard its fully connected (FC) layers to keep only the sub-network. To more robust to spatial translations of the input, the sub-network is followed by the global average pooling layer. Although the localization task (task F and P) and object perception (task T) share some features in the feature extraction stage of the backbone, to separate them, we design two main output branches here. This first branch is to complete the task T classification, and we add one 1024-node FC layer and two 512-node FC layers, finally, the object classification results are output through softmax activation function. Since there is a certain degree of correlation between tactile sensing parts P and F, the output of their classes is on the same branch. We add one 1024-node FC layer and one softmax activation function for the output of task P, and then we continues to add one 512-node FC layer and one softmax activation function after the 1024-node FC layer for the output of task F.

To reduce the dependence of O^{2}VNet on the grayscale information and make MechTac more robust to changing environments, we propose one new binarized convolutional (BConv) layer, and it is added in front of the backbone of O^{2}VNet.

B. Binarized Convolution

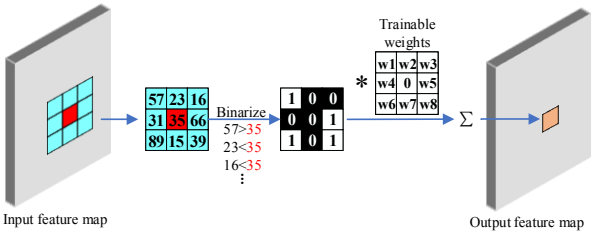


Fig. 6. Binarized Convolution

The standard 2D convolution operation is mainly composed of two parts. First, the input feature map is sampled by k×k convolution kernels, and then the sampled values are weighted and eventually summed and fused. Let us take k=3 as an example, and the standard 3×3 convolution operation is defined as

$$\mathbf{Conv}(x, y) = \sum_{dx=}^{1} \sum_{dy=}^{1} \omega(dx, dy)\, \mathbf{I}(x + dx, y + dy) \quad (2)$$

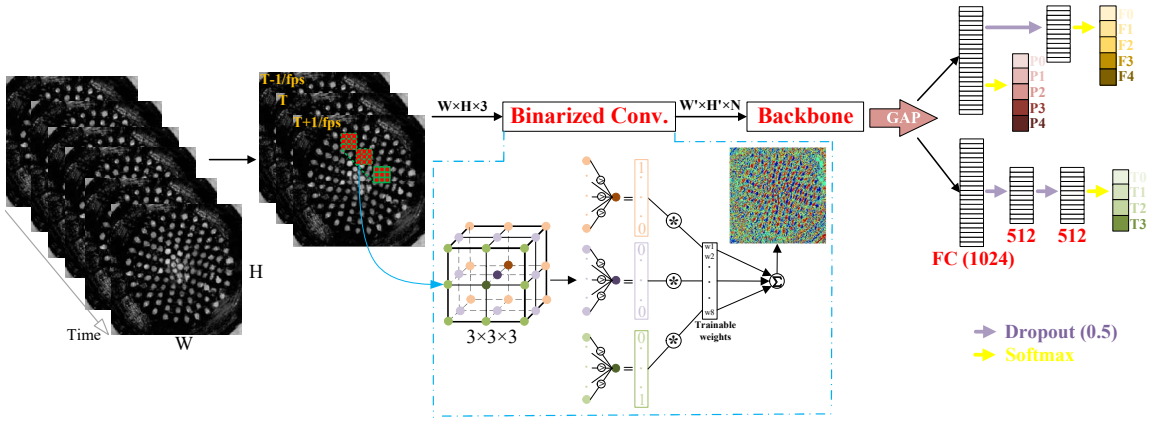


Fig. 7. Overview of network architecture, called perceiving out-of-view neural network (O^{2}VNet). The proposed binarized convolutional (BConv) layer can reduce the effect of image contrast changes, and the O^{2}VNet can separate the weak information (classification task F and P) from the mixed information by two main branches including some FC layers and softmax activation functions.

where **Conv** $(\cdot)$ is the feature maps after convolution operation. **I** $(\cdot)$ denotes the original feature maps. x and y represent the location of the pixel in the image coordinate system. $\omega\,(dx, dy)$ denotes the weight of convolution kernel. Each position of the convolution kernel is designed by $-1 \leq dx \leq 1$ and $-1 \leq dy \leq 1$.

According to Eq. 2, the standard convolution operation relies too much on the grayscale information of the image. Inspired by the grayscale invariance advantage of Local Binary Pattern (LBP) [12], the grayscale images are transformed into images of integer labels to reduce the O^{2}VNet's dependence on grayscale information. The BConv has the same convolution kernel as the standard convolution. However, before the convolution operation, a series of operations need to be performed on the convolution area for the BConv. The feature maps and convolution in CNNs are 3D, so the operation remains the same across the channel dimension. We will model the BConv based on a 3×3 convolution kernel (see Fig. 6).

$$\mathbf{BConv}\,(x, y) = \sum_{dx=\ 1}^{1} \sum_{dy=\ 1}^{1} \omega\,(dx, dy)\, S(\sigma\,(dx, dy)) \quad (3)$$

$$\sigma\,(dx, dy) = \mathbf{I}\,(x + dx, y + dy) - \mathbf{I}\,(x, y) \quad (4)$$

$$S(\sigma\,(dx, dy)) = \begin{cases} 1, & \sigma\,(dx, dy) \geq 0 \\ 0, & \sigma\,(dx, dy) < 0 \end{cases} \quad (5)$$

where $S(\cdot)$ denotes the binarization method. The sensitivity of O^{2}VNet to gray value variations is reduced by it.

C. Loss Function

The total loss l of O^{2}VNet is mainly composed of three parts: task T loss function l_T, task F loss function l_F, and task P loss function l_P. Hence, $l = \lambda l_T + \alpha l_F + \beta l_P$, where λ, α and β are the hyper-parameters (here chosen equal to 1) to balance the different losses.

For each task, we use the softmax version of focal loss functionn [17], because it addresses class imbalance better than the categorical cross-entropy, and can obtain a faster convergence for the multi-classification task:

$$FL = \sum_{c=1}^{m} -\delta(1 - p_c)^{\gamma} g_c \log(p_c) \quad (6)$$

where m denotes number of classes, c denotes class. g_c denotes the ground truth of class c. p_c denotes the prediction results of class c from the softmax. γ is equal to 2, but δ is different for different classification tasks. For the task T, the δ is equal to [0.25, 0.25, 0.25, 0.25]. For the task F and P, the δ is equal to [0.25, 0.25, 0.25, 0.25, 0.25].

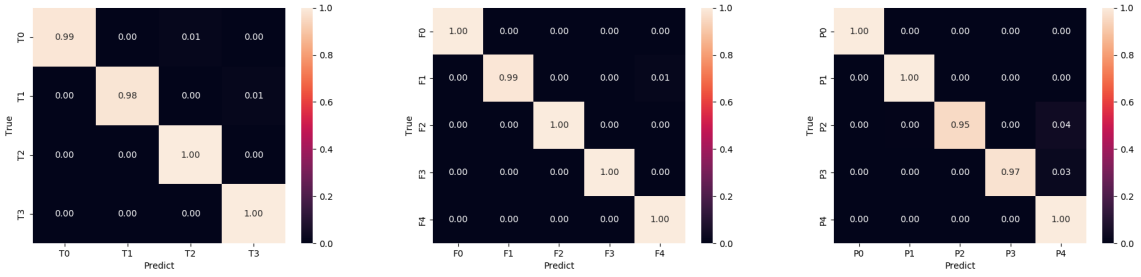
D. Implementation and Experimental Setup

We trained O^{2}VNet on Keras/TensorFlow using a NVidia Quadro P6000 GPU. We used the Adam optimizer (lr = 1e-5) and did not use learning rate decay. We trained O^{2}VNet with a maximal number of 200 epochs and a batch size of 16, and we used early stopping to stop the training when the metric are not optimized anymore on the validation set. The image input to the O^{2}VNet was preprocessed by subtracting the mean.

To reduce overfitting of the O^{2}VNet, we used some methods such as the regularization technique and data augmentation. We added a dropout layer of 0.5 rate and one L2 regularization after each FC layer. For the data augmentation, it included that randomly rotated images in the range ±1 degree, randomly zoomed images in the range ±20%, randomly shifted images horizontally ±0.1 of the total width W of the image, and randomly shifted images vertically ±0.1 of the total height H.

E. Evaluation Methods

The classification accuracy and the confusion matrix [18] are used to evaluate our classification results. The classification accuracy is the ratio between the number of correct predictions and the total number of predictions made. The confusion matrix is a cross-tab, with rows and columns representing the true and predicted classification, and the main diagonal representing the correctly classified elements.



Task T Task F Task P

Fig. 8. Confusion matrix of each task. Task T denotes object classification (T0 to T3) on the TacTip. Task F and P denote the localization of touching regions (rough/precise localization) on the TacSide.

IV. Experiments

A. Results in Weak Signal Separation

To verify the effectiveness of the O^{2}VNet, we first train and validate it on the **TAC** dataset. Then, after obtaining the optimal classification model, we evaluate it on the test set of **TAC**. As shown in Fig. 8, we obtain the confusion matrix of each task on the test set. High classification accuracy for each classification task is very important for the localization of touching regions and object perception. For the object perception, we simplify the objects (see Fig. 4) to sphere (T1), concave (T2), prismatic (T3), and the combination of these three objects can represent the situation where most objects are in contact with the TacTip. The classification accuracy of O^{2}VNet for each class of P is close to 99%. For the localization of touching regions, the classification of task F is easier than the task P, because its touch area is larger than task P (see Fig. 3), it is easier to distinguish. Since the TacSide's touch region information is transmitted to the TacTip through tendons (see Fig. 2), and the tendons are not arranged all over the TacSide, which makes the signal strength transmitted to the TacTip relatively weak. The regions from P1 to P4 are adjacent, misclassifications tend to occur in adjacent regions between them, so for the task P, there are more misclassifications than the task F. **But the results show that the O^{2}VNet successfully separates the weak information (localization task) from the mixed information based on the monocular images**.

TABLE I
Classification accuracy/% of O^{2}VNet with/without BConv

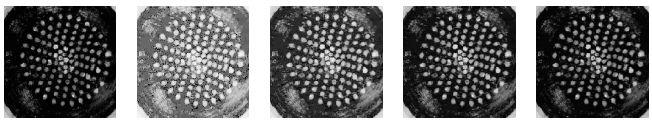
BConv	Class			Parameters
	T	F	P	
✗	98.92	99.69	**99.11**	10,458,702 (∼10M)
✔	**99.22**	**99.71**	98.24	10,458,785 (∼10M)

B. Results in Contrast Image Changes

We conduct one comparative experiment with/without the BConv layer, as shown in Tab. I. The classification accuracy of our method with/without BConv on the test set is similar, so it is not easy to say which method with/without BConv is better. Therefore, we further test it and perform a perturbation experiment on the test set of **TAC**. Captured images are susceptible to light intensity. Hence, to test the performance of O^{2}VNet on contrast variations, we use the contrast function (Eq. 7) of image augmentation tool [19] to change the contrast of the captured images.

$$I_{contrast}(x,y) = 255 \times (\frac{I(x,y)}{255})^{\eta} \tag{7}$$

As shown in Fig 9, the image contrast is changed by the η in Eq. 7, and we set the η to 0.2, 0.4, 0.6, and 0.8, respectively.



Original image η=0.2 η=0.4 η=0.6 η=0.8

Fig. 9. Contrast images for different η in Eq. 7.

In the Tab. II, the classification accuracy of O^{2}VNet with BConv does not decrease even with contrast changes. The O^{2}VNet has good stability to contrast changes without significantly increasing parameters (only added 83 parameters). Therefore, **the O^{2}VNet with BConv layer makes MechTac more robust to changing environments.**

TABLE II
Classification accuracy/% of O^{2}VNet with/without BConv for perturbation experiment

BConv	η	Class		
		T	F	P
✗	0.2	24.90	41.37	26.67
	0.4	46.47	57.06	35.78
	0.6	64.90	89.22	56.67
	0.8	88.04	98.63	92.55
✔	0.2	99.12	99.71	98.33
	0.4	99.02	99.71	98.33
	0.6	99.12	99.71	98.14
	0.8	99.12	99.71	98.24

C. Results in Different Backbones and Input Channels

When choosing the backbone of O^{2}VNet, we compare the results of various backbones (see Tab. III). Keeping the entire network framework in Fig. 7 unchanged, we only replace different backbones to train O^{2}VNet on the **TAC** dataset. According to the number of images in **TAC** dataset, it is not as large as the ImageNet dataset [16], and large classification models are prone to overfitting, so small models are preferred. DenseNet121 [10] serves as the backbone of O^{2}VNet, allowing the O^{2}VNet to achieve good

TABLE III

CLASSIFICATION ACCURACY/% OF DIFFERENT BACKBONES

Backbone	Class			Parameters
	T	F	P	
DenseNet121 [10]	**99.22**	**99.71**	**98.24**	10,458,785 (∼10M)
VGG16 [20]	73.92	99.31	90.59	17,087,393 (∼17M)
Xception [21]	84.61	97.45	91.67	26,379,913 (∼26M)
InceptionV3 [22]	92.55	99.51	97.16	27,321,217 (∼27M)
ResNet50 [23]	65.49	91.76	83.04	29,106,145 (∼29M)
ResNet101 [23]	84.12	93.73	95.00	48,176,609 (∼48M)

classification accuracy on each task. However, under the existing network framework, other backbones only perform well on a particular classification task and cannot take into account each task.

TABLE IV

CLASSIFICATION ACCURACY/% OF DIFFERENT INPUT CHANNELS N

N	Class			Parameters	Time/ms
	T	F	P		
3	**99.22**	**99.71**	**98.24**	10,458,785 (∼10M)	34
5	80.49	97.35	97.18	10,458,833 (∼10M)	36
7	40.88	91.08	64.71	10,458,881 (∼10M)	37

The size of our O^{2}VNet model is 167MB. We test the different input channels of O^{2}VNet (see Tab. IV). In this paper, we input three consecutive images (T-1/fps, T, T+1/fps) into O^{2}VNet, and its classification result is the best. When we input five or seven consecutive images, the classification results of O^{2}VNet drop sharply, and the reason for this drop is related to frame per second (fps). The fps directly affects the continuity of movement. In the process of collecting data, the fps is small. Hence, with the larger the number of input continuous images, the continuity of changes is more affected. So in the face of small fps, choosing a small number of input continuous images is more beneficial to the classification accuracy of the O^{2}VNet. The computation time of the entire pipeline is 34 ms for one image, making it usable for MechTac.

V. CONCLUSION

In this paper, we design one vision-based tactile sensor, MechTac, which combines mechanical transmission and optical detection to increase the tactile regions. To make it have good perception ability, we propose one perceiving out-of-view neural network, O^{2}VNet, taking deformation images from the monocular camera of MechTac as input to localization of touching regions and objects perception. We prepare one tactile image dataset, **TAC**, and each image is labelled with touch regions and object categories. O^{2}VNet shows a superior classification accuracy on test set of **TAC**, which not only proves the effectiveness of O^{2}VNet to multi-functional classification task, but also has the ability to separate specific information from mixed information. And facing to variation of light intensity, O^{2}VNet still maintains high classification accuracy due to the proposed BConv layer. Finally, the classification results shows the enormous potential of MechTac for robot manipulation.

REFERENCES

[1] N. F. Lepora, "Soft biomimetic optical tactile sensing with the tactip: A review," *IEEE Sensors J.*, 2021.
[2] S. Wang, M. M. Lambeta, P.-W. Chou, *et al.*, "Tacto: A fast, flexible, and open-source simulator for high-resolution vision-based tactile sensors," *IEEE Robotics and Automation Letters*, 2022.
[3] B. Ward-Cherrier, N. Pestell, L. Cramphorn, *et al.*, "The tactip family: Soft optical tactile sensors with 3d-printed biomimetic morphologies," *Soft robotics*, vol. 5, no. 2, pp. 216–227, 2018.
[4] Y. Zhang, Z. Kan, Y. Yang, *et al.*, "Effective estimation of contact force and torque for vision-based tactile sensors with helmholtz–hodge decomposition," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4094–4101, 2019.
[5] 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.
[6] Y. Zhang, Z. Kan, Y. A. Tse, *et al.*, "Fingervision tactile sensor design and slip detection using convolutional lstm network," *arXiv preprint arXiv:1810.02653*, 2018.
[7] A. Padmanabha, F. Ebert, S. Tian, *et al.*, "Omnitact: A multi-directional high-resolution touch sensor," in *2020 IEEE International Conference on Robotics and Automation*. IEEE, 2020, pp. 618–624.
[8] C. Trueeb, C. Sferrazza, and R. D'Andrea, "Towards vision-based robotic skins: a data-driven, multi-camera tactile sensor," in *2020 3rd IEEE International Conference on Soft Robotics*. IEEE, 2020, pp. 333–338.
[9] C. Choi, W. Schwarting, J. DelPreto, *et al.*, "Learning object grasping for soft robot hands," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2370–2377, 2018.
[10] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 4700–4708.
[11] 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.
[12] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, 2002.
[13] G. Wang, W. Li, S. Ourselin, *et al.*, "Automatic brain tumor segmentation based on cascaded convolutional neural networks with uncertainty estimation," *Frontiers in computational neuroscience*, vol. 13, p. 56, 2019.
[14] Z. Zhao, E. Puybareau, N. Boutry, *et al.*, "Do not treat boundaries and regions differently: An example on heart left atrial segmentation," in *2020 25th Int. Conf. Pattern Recog.* IEEE, 2021, pp. 7447–7453.
[15] N. F. Lepora, A. Church, C. De Kerckhove, *et al.*, "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.
[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Adv. Neural Inform. Process. Syst.*, vol. 25, 2012.
[17] T.-Y. Lin, P. Goyal, R. Girshick, *et al.*, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.
[18] M. Grandini, E. Bagli, and G. Visani, "Metrics for multi-class classification: an overview," *arXiv preprint arXiv:2008.05756*, 2020.
[19] A. B. Jung, K. Wada, J. Crall, *et al.*, "imgaug," https://github.com/aleju/imgaug, 2020, online; accessed 01-Feb-2020.
[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
[21] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 1251–1258.
[22] C. Szegedy, V. Vanhoucke, S. Ioffe, *et al.*, "Rethinking the inception architecture for computer vision," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 2818–2826.
[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 770–778.