2019 International Conference on Robotics and Automation (ICRA)
Palais des congres de Montreal, Montreal, Canada, May 20-24, 2019

Morphology-Specific Convolutional Neural Networks for Tactile Object Recognition with a Multi-Fingered Hand

Satoshi Funabashi, Gang Yan, Andreas Geier, Alexander Schmitz, Tetsuya Ogata and Shigeki Sugano

Abstract— **Distributed tactile sensors on multi-fingered hands can provide high-dimensional information for grasping objects, but it is not clear how to optimally process such abundant tactile information. The current paper explores the possibility of using a morphology-specific convolutional neural network (MS-CNN). uSkin tactile sensors are mounted on an Allegro Hand, which provides 720 force measurements (15 patches of uSkin modules with 16 triaxial force sensors each) in addition to 16 joint angle measurements. Consecutive layers in the CNN get input from parts of one finger segment, one finger, and the whole hand. Since the sensors give 3D (x, y, z) vector tactile information, inputs with 3 channels (x, y and z) are used in the first layer, based on the idea of such inputs for RGB images from cameras. Overall, the layers are combined, resulting in the building of a tactile map based on the relative position of the tactile sensors on the hand. Seven different combination variations were evaluated, and an over-95% object recognition rate with 20 objects was achieved, even though only one random time instance from a repeated squeezing motion of an object in an unknown pose within the hand was used as input.**

I. INTRODUCTION

Tactile sensing in robot hands is complementary to other sensing modalities, for example in the case of visual occlusions or for object qualities like softness and adhesiveness. Especially when using multi-fingered hands, diverse and relatively large areas (i.e., not only the fingertips) are in contact with the object, and forces are exerted in various directions. The processing of such abundant tactile information is challenging. While the current paper focuses on tactile object recognition, we suggest that the insights and general idea (i.e. MS-CNN) can be applied to various tasks.

A lot of research has been conducted into how robotic hands and tactile sensors achieve tasks [1]. Considering grasping states and positions of fingertips with analytical solutions, in-hand manipulation or changing the desired states of objects can be robustly executed [2][3]. However, complicated grasping states make analytical modeling difficult, such that often only two fingered hands are used. Not only tactile sensors but also cameras are used for control [4]. The hand has to be a simple configuration so that it can avoid occlusion, but this limits the ability to achieve the difficult tasks that multi-fingered hands can do. Other research uses machine learning to achieve various tasks [5], but training with high-dimensional tactile information is challenging.

This research was supported by the Japan Science and Technology Agency ACT-I Information and Future No. 50185. *(Corresponding author: Satoshi Funabashi)*

The authors are with the Faculty of Science and Engineering, Dept. of Modern Mechanical Engineering, Waseda University, Tokyo 169-8555, Japan. (e-mail: s_funabashi@sugano.mech.waseda.ac.jp)

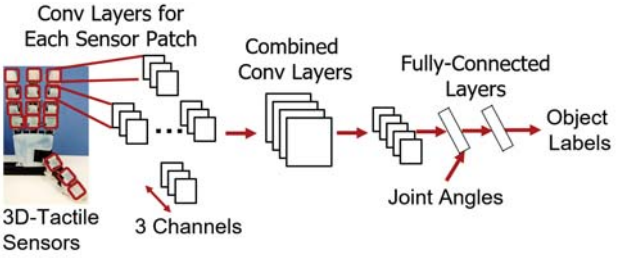

Fig. 1: Proposed convolutional neural network.

Some of the latest research dealing with robotic hands and abundant tactile information make use of deep neural networks. Previously, our group also used deep learning with the TWENDY-ONE Hand for object recognition and in-hand manipulation [6][7] and achieved improved results to other modeling and machine learning methods. A stacked autoencoder was used for the tasks. Even though the hand has distributed tactile sensors, the geometrical information of the tactile sensors was not considered for the architecture of the stacked autoencoder. In some of the research into distributed tactile sensors, convolutional neural networks (CNNs) are preferably used. A number of researchers have worked on CNNs for image and speech recognition because CNNs are robust to the extraction of features from two-dimensional information. That advantage of CNNs also matches distributed tactile sensors. Even though CNNs have produced good results, when it comes to multi-fingered hands, the question of how to input the tactile information to a CNN is still challenging. In particular, the size and shape of the tactile patches on the hand varies, as well as the size of the fingers, which makes the implementation of CNNs difficult, as CNNs in general require rectangular-shaped inputs.

In this paper, we use the uSkin 3D tactile sensors [8][9] attached to the Allegro Hand, and we use the same dataset as in our previous paper [10]. However, the previous work only used simple neural networks and did not take the geometric relations of the sensors into account. Only when using time-series information as the input could tactile object recognition rates of 95% be achieved, while the current paper achieves comparable recognition rates when using only a single random time instance as input. As part of the MS-CNN, the current paper implements the following:

- Filters for the MS-CNN take as input sensor readings from a part of a finger segment, a whole finger segment, a whole finger, and the whole hand, in a consecutive manner. However, certain differences are addressed, i.e., the fingertips have a different morphology than the other

978-1-5386-6027-0/19/$31.00 ©2019 IEEE

finger segments, and the thumb has a different layout than the other fingers.
- uSkin provides distributed 3-axis tactile sensing. Therefore, in the input layer, channels for the x, y, and z axes are used.

Object recognition is targeted as a first step among multi-fingered hand tasks. A new CNN architecture is proposed (Fig. 1); this paper focuses on how to combine convolution layers and which combination of layers provides the highest object recognition rate. Moreover, object recognition rates with 3D and 1D tactile information are investigated and compared.

II. Related Works

A. Machine Learning and Recognition Tasks

A variety of machine learning methods are used to accomplish recognition tasks. Random Forest has been used for object classification with a simple two-fingered robotic hand [11]. Transfer learning has also been used for tactile sensing tasks and was shown to achieve a high recognition rate with a small training dataset [12]. In this research, the processing of training data is limited to applications on flat surfaces only. A Support Vector Machine (SVM) can also be used for zero-shot learning of object recognition tasks, although the tactile information is limited to the fingertips because this kind of machine learning method is less capable of dealing with high-dimensional tactile information [13]. Other researchers have attempted a fast estimation of object shape recognition in which the tactile sensor is only on one fingertip [14]. Another active exploration method used SVM and combined tactile and visual information for training, the accuracy of which was worse than training with either one on its own [15]. Some research has focused on distributed tactile sensors [16][17]. When used with SVM and SOM (self-organized map), the recognition rate is still low. Other research has already focused on the traits of a multi-fingered hand [18]. High recognition rates for objects can be achieved by concatenating the information from each finger. Although a lot of research has already been conducted on the combination of robotic hands and tactile sensors, it remains challenging to deal with large numbers of tactile sensors. In short, there are limitations of machine learning when it comes to processing too high-dimensional tactile information.

B. Convolutional Neural Network

As one of the methods used to process abundant tactile information, deep learning is used for tactile sensors. Using stacked autoencoders, in-hand manipulation with differently sized and shaped objects can be achieved [7]. Deep reinforcement learning is also used and has accomplished the changing of orientation of a cylindrical object; however, the hand is multi-fingered, which could make situations difficult because of the many degrees of freedom of the fingers' joints [19]. Furthermore, CNNs have been widely utilized for distributed tactile sensors on robotic hands [20][21][22][23][24]. Texture recognition and slip detection have also been achieved and CNNs get better recognition rates in comparison with other machine learning methods like, for example, SVM [25]. In these cases, the CNNs are trained from the viewpoint for positions of tactile sensors like image processing. The state of the art is focused on CNNs and distributed tactile sensors [26][27]. When it comes to multi-fingered hands, they have differently sized and shaped distributed sensors [28][29], which means that the question of how to input tactile information from such sensors to CNNs needs to be considered; this question has not yet been investigated.

III. Proposed Method

A. Input mapping from 3D tactile sensors

The positions of sensors on phalanges and fingertips are described in Fig. 2. The number of sensors on each sensor patch is 16. However, the size and shape of phalanges and fingertips are different from each other. In terms of the positions of sensors, the shapes of the input maps for phalanges are 4 x 4 and for fingertips are 6 x 4. For the input maps for fingertips, the number "0" (red number in Fig. 2) is input to positions where no sensors are mounted on each fingertip, resulting in rectangular input maps. This enables that the input maps are convoluted by filters with a size of 2 x 2 or more.

Some image recognition research has employed inputs with three channels for "RGB" because each pixel on an image has "RGB" information. From this point, in the current paper, inputs with three channels to CNNs are used because each sensor (or "taxel") provides "xyz" information (described in Fig. 2), and Fig. 4 shows the direction of the xyz information on a patch. For the fingertips, the direction of xyz changes along with the curve of the fingertips. In particular, the z-axis is the component of the force normal to the sensing surface.

B. Combining convolution layers

Multi-fingered hands could have differently sized and shaped tactile sensors on the hands. In our case, for the Allegro Hand, the size and shape of tactile patches are different and also the number of tactile patches on the index, middle, and little fingers is four, while that of the thumb is three. One method of dealing with these tactile sensor

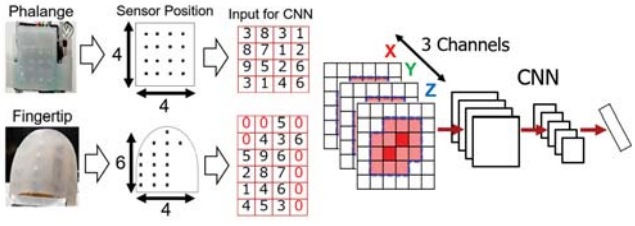

Fig. 2: From the left side, mounted sensor patches, black dots as positions of sensors, maps for inputting to CNN. The red-colored '0' represents the position where sensors are not mounted on actual sensor patches (the other numbers are arbitrary). The maps are input with 3 channels (x, y, z) to a CNN.

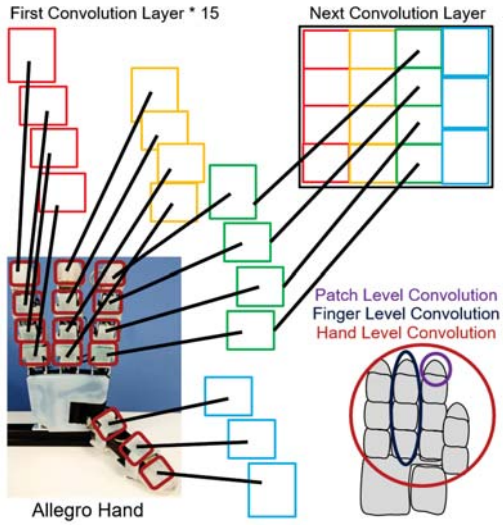

Fig. 3: One example of how to combine convolution layers from each of the tactile sensor patches. For our current robotic hand platform, the size of the sensor patches differs between the fingertips and phalanges. Furthermore, the number of patches is different between the thumb and the other fingers. In the example presented in this figure, convolution layers are prepared for each tactile patch (Patch Level Convolution) and combined in a later stage. Further details are presented in Table I.

patches is described in Fig. 3. In this case, convolution layers for each tactile patch are prepared as the first convolution layer. In the next layer, all the convolution layers in the first layer are combined by following the positions of tactile sensors on the hand. In order to make the layer's shape rectangular for filtering, the inputs from each patch for the first layer are convoluted to transform into outputs with exact sizes for the next layer. The sizes (heights and widths) of the filters for the convolution layers in the first layer are adjusted as follows:

$$OH = \frac{H + 2P - FH}{S} + 1$$
$$OW = \frac{W + 2P - FW}{S} + 1 \quad (1)$$

where H and W are the height and width of the inputs for the current convolution layer; OH and OW are the height and width of the outputs for the current convolution layer; FH and FW are the height and width of the filters for convoluting the inputs to make outputs for the next convolution layer; P is a padding, to which usually '0' is added around the input maps so that the size of outputs from the convolution layer would be same as that of the outputs for the next convolution layer; finally, S is the size of a stride which the filters shift on inputs. Padding is not used in this paper because the convolution layers change the size of inputs for combining the convolution layers. Also, we focus on how to combine convolution layers; thus, the CNNs used in our experiments are without pooling layers which can also change the size of inputs.

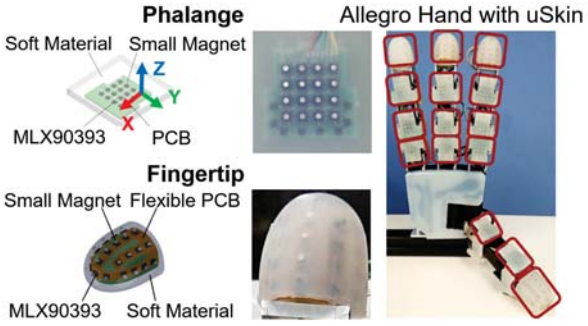

Fig. 4: Allegro Hand with uSkin 3D tactile sensors for phalanges and fingertips. Four uSkin sensor patches are mounted on the index, middle, and little fingers, and three patches are mounted on the thumb.

IV. Experimental Design

A. Allegro Hand with uSkin

In this research, an Allegro Hand is used, which is a commercially available robotic hand from Wonik Robotics. In previous research by our lab, the uSkin tactile sensors, which can detect triaxial force measurements, were presented for covering the phalanges [8] and fingertips [9] of the Allegro Hand, as described in Fig. 4. As the Allegro Hand is a multi-fingered robotic hand with 16 degrees of freedom, forces with diverse orientations could occur during task execution. Therefore, uSkin sensors were implemented to detect such complicated grasping or manipulating states. A total of 15 uSkin patches were installed, covering all the phalanges and the fingertips of the Allegro Hand (4 patches on the index, middle and little fingers, 3 patches on the thumb). uSkin patches have not yet been mounted on the palm of the hand. In total, our customized Allegro Hand can provide 16 (joint angles) + 15 (sensor patches) x 16 (sensors) x 3 (tactile axes) = 736 measurements.

B. Combining Patterns

A variety of patterns were investigated for combining the convolution (Conv) layers. The positions of the sensors on the hand were taken into consideration when defining those patterns. In particular, whether to apply filters across the boundaries of finger segments and fingers in different stages was investigated. There are 7 patterns, as shown in Fig. 5, and the parameters for each pattern are presented in Table I. For example, Pattern I has inputs with a size of 18 x 16 x 3 for the first Conv layer. After passing them to the first Conv layer, the size of the outputs from the layer is 10 x 8 x 14. Since the comparison of 3D and 1D tactile information is to be evaluated, there are parameter settings for both 3D (in bold letters) and 1D (since we want to compare with standard tactile sensors which provide normal force, only the z axis is used as it represents normal force from the uSkin), which are heuristically determined. Patterns II, III, and IV have two filter size settings for one of the Conv layers because of the adjustment for the size of Conv layers explained in the

description of each pattern below. Joint information is added to the 1st FC layer represented by 'J' in Fig. 5 in all the patterns.

Weight sharing can be used to reduce the number of filters that need to be learned. Evaluations with or without weight sharing were performed. In the case of weight-sharing, the same set of filters is used for inputs of the same dimension in a convolution layer (in a Patch or Finger Map Layer). For example in the 'Patch Map Layer', filters with the same weights are used for each patch. To adjust the size of convolution layers, different sized filters are sometimes used (in Patterns II, III, and IV) and the same sized filters share the same weights.

In Pattern I, the 'Hand Map Layer with Zero Padding' is constructed as the input layer with a size of 18(rows) x 16(columns). Three tactile patches are mounted on the thumb (4 x 4 x (2 patches) and 6 x 4 x (1 patch)), while the other fingers have four patches (4 x 4 x (3 patches) and 6 x 4 x (1 patch)) each. Therefore, to construct a rectangular input layer, 4(rows) x 4(columns) of '0' are added above the input from the thumb's fingertip, as described in Fig. 5. As a consequence, the filters get applied across the boundaries of the finger segments already in the first layer. This pattern is for confirming whether combining all information into one big input layer is sufficient.

For Pattern II, the 15 sensor patches are processed individually in the first convolution layer (Patch Map Layer). This means that the filters are not applied across the boundaries of sensor patches, and different filters are trained for different sensor patches, unless weight sharing is used. For the next layer, the output of the first convolution layer is combined by considering the positions of tactile patches on the hand described in Fig. 3 as the Hand Map Layer. Filters with a size of (2,3) in the 1st Conv layer are used for input maps from the phalanges on the thumb. Details are presented in Table I.

Pattern III uses a Patch Map Layer also for the 2nd Conv layer. The 'Hand Map Layer' is implemented in the 3rd Conv layer. Filters with a size of (2,2) are used in the 2nd Conv layer for the fingertip of the thumb. The difference between Pattern II and III lies in when to build the 'Hand Map Layer'. This idea is derived from [30] which investigated when to fuse information for grasp stability.

In Pattern IV has a 'Patch Map Layer' in the input layer and 1st Conv layer. The combining of convolution layers is executed in the 2nd Conv layer, which results in the construction of four finger maps (index, middle, little, and thumb) as the 'Finger Map Layer'. The 'Hand Map Layer' is constructed in the 3rd Conv layer. This pattern is to change map patterns from the 'Patch Map Layer' to the 'Hand Map Layer' by one Conv layer at a time. In Table I, filters with a size of (4,2) in the 2nd Conv layer are used for a convolution layer from the phalanges and fingertip of the thumb.

Pattern V uses the 'Patch Map Layer' in the input layer and 1st Conv layer. The 'Finger Map Layer' is constructed in the 2nd and 3rd Conv layers. Pattern V gradually combines convolution layers without a 'Hand Map Layer', unlike Pattern IV.

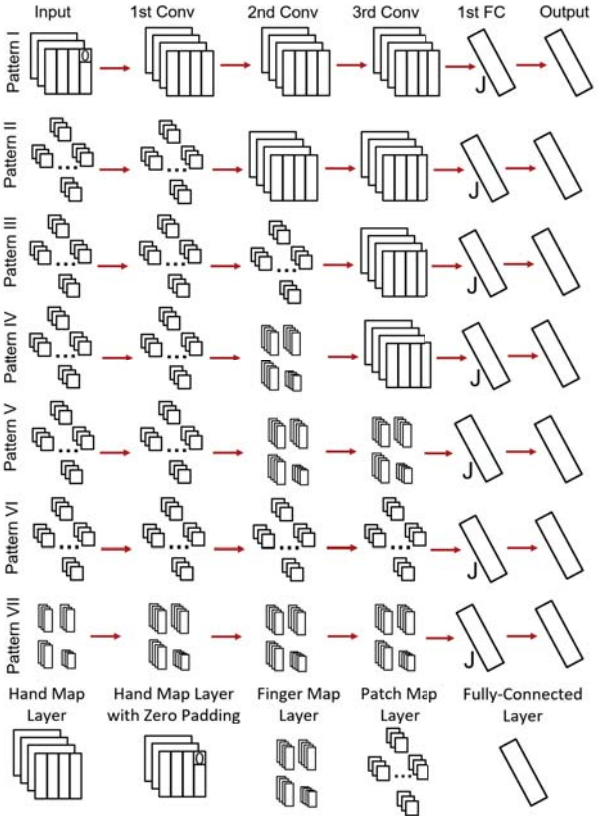

Fig. 5: Seven patterns for combining convolution layers. The 1st FC gets features from the 3rd Conv layer and joint angles (represented by 'J').

Pattern VI has only the 'Patch Map Layer' in the input and all the Conv layers. This pattern is for confirming if combining according to positions of patches on the hand is necessary.

Pattern VII only uses the finger 'Finger Map Layer' in the input and all the Conv layers so that we confirm if only considering the positions of patches on the fingers is enough for recognition.

C. Training Setting

The CNNs used in this paper have the same network parameters and training settings except for the size of the convolution layers described in Table I. The CNNs are trained with 12,500 samples for up to 20,000 epochs and 2,500 different samples are used as the test set. Relu was used as an activation function for all the layers except the output layer. The output layer has a Softmax activation function. For all the CNNs, all the weights in each network are initialized by random numbers from a trancated normal distribution with a standard deviation of 0.1 and the random seed was defined as 1. The learning rate was set to 0.00001, Adam was used as the optimizer, and the minibatch size was 100. For the optimizer, the step size α was 0.0001, the first exponential decay rate $\beta 1$ was 0.9, the second exponential decay rate $\beta 2$ was 0.999 and the small value for numerical stability ε was 1e-08. All the CNNs were built with the

TABLE I: Network Setting

		1st Conv	2nd Conv	3rd Conv	1st FC	Output
Pattern I	In	**3**, 1	**14**, 5	**28**, 9	**1136**, 376	**1200**, 400
	Out	**14**, 5	**28**, 9	**56**, 18	**1200**, 400	20
	Filter Size	(3,3)	(3,3)	(2,2)	-	-
	Stride	(2,2)	(1,1)	(1,1)	-	-
Pattern II	In	**3**, 1	**14**, 5	**28**, 9	**1136**, 376	**1200**, 400
	Out	**14**, 5	**28**, 9	**56**, 18	**1200**, 400	20
	Filter Size	(3,3)/(2,3)	(3,3)	(4,3)	-	-
	Stride	(1,1)	(1,1)	(1,1)	-	-
Pattern III	In	**3** ,1	**14**, 5	**28**, 9	**912**, 304	**1200**, 400
	Out	**14**, 5	**28**, 9	**56**, 18	**1200**, 400	20
	Filter Size	(3,3)	(2,2)/(3,2)	(2,1)	-	-
	Stride	(1,1)	(1,1)	(1,1)	-	-
Pattern IV	In	**3**, 1	**14**, 5	**28**, 9	**1024**, 340	**1200**, 400
	Out	**14**, 5	**28**, 9	**56**, 18	**1200**, 400	20
	Filter Size	(3,3)	(2,2)/(4,2)	(2,2)	-	-
	Stride	(1,1)	(1,1)	(1,1)	-	-
Pattern V	In	**3**, 1	**14**, 5	**28**, 9	**1192**, 394	**1200**, 400
	Out	**14**, 5	**28**, 9	**56**, 18	**1200**, 400	20
	Filter Size	(2,2)	(5,2)	(5,2)	-	-
	Stride	(1,1)	(1,1)	(1,1)	-	-
Pattern VI	In	**3**, 1	**14**, 5	**28**, 9	**1304**, 430	**1200**, 400
	Out	**14**, 5	**28**, 9	**56**, 18	**1200**, 400	20
	Filter Size	(2,2)	(2,2)	(2,2)	-	-
	Stride	(1,1)	(1,1)	(1,1)	-	-
Pattern VII	In	**3**, 1	**14**, 5	**28**, 9	**1024**, 340	**1200**, 400
	Out	**14**, 5	**28**, 9	**56**, 18	**1200**, 400	20
	Filter Size	(4,2)	(3,2)	(2,1)	-	-
	Stride	(2,2)	(1,1)	(1,1)	-	-

Tensorflow library for Python and a GTX Geforce 1080 was used as the GPU.

D. Data Collecting

The training data used are the same as those in our previous paper [10], in which we confirmed our skin sensors' effect with a multi-fingered hand by focusing on time-series information to get better accuracy in object recognition with neural networks. In the current paper, we focus on the positions of tactile sensors and patches (morphology) on the Allegro Hand with same training data so that we could confirm the effects of time-series and morphology of the hand independently. The data collection setting is only briefly described here because the detail has been provided in [10]. The dataset includes the tactile information from a series of in-hand manipulations as an active tactile sensing method. Presented in Fig. 6 are the 20 common objects, including 10 objects from the Yale-CMU-Berkeley Model Set [31], that were used for our experiment. Thirty manipulation trials for each object were conducted, resulting in 600 trials overall. Twenty-five trials for each object were randomly selected for training a CNN, while five trials for each object were used for the test set for the respective CNN. The objects were placed on the Allegro Hand in random positions. Data was collected with a sampling rate of 30 Hz, and 250 time steps for each trial were collected. Twenty-five time steps were randomly

Fig. 6: Target objects and training data collection. The top image shows target objects; ten of the objects were selected from the Yale-CMU-Berkeley Model Set and the others are common objects which have similar shapes to each other. The bottom row images show how the training data were collected.

sampled from 250 time steps for each trial as a training dataset. In total, the training dataset included 12,500 samples. The test dataset consisted of samples randomly sampled from 5 trials for each object resulting in 2,500 samples. All the collected samples included the 16(joint angles) + 15(sensor patches) x 16(sensors) x 3(forces axes) resulting in 736 measurements at each time step from the Allegro Hand. When training the CNNs, there are two ways to add '0' to the inputs. As described in Section III-A, the number '0' is added to the inputs from sensors on the fingertips for providing to CNNs. As a result, the number of dimensions for the inputs is 736 + 8(the number of '0' for one fingertip) x 3(forces axes) x 4(number of fingertips) = 832 dimensions of inputs. The other way is as described in Section IV-B, wherein 4(rows) x 4(columns) of '0' is added to the inputs to Pattern I. As a result, the number of dimensions for the inputs is 736 + 8(the number of '0' for one fingertip) x 3(forces axes) x 4(number of fingertips) + 16(4(rows) x 4(columns) for above the fingertip of the thumb) x 3(forces axes) = 880 dimensions of inputs for Pattern I.

V. Evaluation

In tables II to IV, the accuracies and variances presented are averages of ten recognition trials. 1,500 samples were randomly selected for each trial from 2,500 samples of test sets.

A. Comparison of Combining Patterns

Table II shows the recognition rates from the seven patterns described in Section IV-B with 1D tactile information. After 8,300 training epochs, Pattern I and II achieved approximately 80% accuracy. The others achieved better accuracy, over 82%. However, the variance for Pattern VI was high at 9.87. Since Pattern VI has a lot of weights (convolution layers for each patch (Patch Map Layer) through all the Conv layers) compared to the others, it was difficult for the

TABLE II: Accuracy of CNNs with 1D

	8,300 epochs		20,000 epochs	
	Accuracy %	Variance	Accuracy %	Variance
Pattern I	80.38	3.91	86.47	3.41
Pattern II	79.80	6.48	87.93	5.15
Pattern III	86.17	4.60	89.87	6.67
Pattern IV	82.86	5.76	87.26	5.23
Pattern V	85.48	4.42	90.93	1.28
Pattern VI	85.57	9.87	91.68	6.34
Pattern VII	84.27	3.83	88.47	2.73

TABLE III: Accuracy of CNNs with 3D after 1,300 epochs

	Different Filters		Shared Filters	
	Accuracy %	Variance	Accuracy %	Variance
Pattern I	81.95	4.10	81.95	4.10
Pattern II	89.76	3.84	88.50	6.15
Pattern III	87.33	3.94	79.85	5.86
Pattern IV	89.45	4.88	81.02	4.89
Pattern V	89.09	7.29	84.28	5.43
Pattern VI	88.69	5.54	79.50	5.73
Pattern VII	81.69	4.88	74.30	7.59

accuracy to converge. After 20,000 training epochs, Pattern V and VI got over 90% recognition rate but Pattern I had 86.47%, the worst recognition rate. The difference between Pattern I and the others is in whether or not the input layer has 'Patch Map Layer'. For Pattern I, we assumed that just making a large hand map input ('Hand Map Layer with Zero Padding') is enough for the recognition. From this result we infer that if the input layer is separated by following the geometry of the hand (e.g., Patch or Finger Map Layer), a higher recognition rate could be achieved.

B. Tactile Dimensionality

By using inputs with 3 channels (x, y, z axes) for 3D tactile information, a comparison of object recognition between CNNs trained with 3D and 1D (z axis) tactile information was conducted. Tables II and III show that the patterns trained with 3D produce better accuracy than patterns trained with 1D. The training with 1D was executed for 20,000 epochs and the accuracies from most of the patterns are similar to those of 3D, while the patterns with 3D are trained for only 1300 epochs. However, patterns I and VII have around an 82% recognition rate, which is lower than the accuracies from patterns with 1D trained for 20,000 epochs. We assume that in addition to using 3D tactile information, the architectures of the CNNs also need to be well-organized. Patterns that have the 'Patch Map Layer' in the input layer could have better accuracies than the other patterns. The possible reason why patterns I and VII could not get better or similar result to the others could be that the sensor patches are not continuously mounted on the hand because of the gaps among the phalanges, fingertips, and fingers, thus it could be hard to apply filters to the boundaries among sensor patches. We confirmed that 3D tactile information is useful from the viewpoint of time-series information in our previous work [10]. This paper also confirmed the usefulness of 3D tactile information from the viewpoint of the geometry of the hand.

TABLE IV: Accuracy of CNNs with 3D after 10,000 epochs

	Different Filters	
	Accuracy %	Variance
Pattern II	93.46	1.26
Pattern III	94.03	1.71
Pattern IV	95.59	0.86
Pattern V	94.01	0.78
Pattern VI	94.13	2.47

C. Combining Patterns and 3D Tactile Information

First, weight-sharing with 3D was evaluated and the results are shown in Table III. Pattern I has the same result because its convolution layers are never separated. For the other patterns, even though Pattern II has the best recognition rate among the patterns with shared filters (weight-sharing), all patterns have lower accuracies with shared filters. The variances for most of the patterns with different filters are lower than with shared filters. Therefore, it is confirmed that different filters should be prepared for different parts of the hand.

Finally, further training was executed for patterns II to VI with different filters which have higher recognition rates after 1,300 epochs. Table IV shows the recognition rates after 10,000 training epochs. Pattern II achieved slightly lower recognition rates compared to the others. The difference between Pattern II and the others is that Pattern II fuses the 'Patch Map Layer' in the 1st Conv layer, whereas the others gradually fuse each convolution layer or do not fuse layers. The best recognition rate is from Pattern IV, which includes the Patch, Finger, and Hand Map layers. Some of the variances are even under 1.00. From these results, we infer that combining convolution layers according to the tactile sensors' configuration on a robot (in this case, a multi-fingered hand) is meaningful for the CNNs, and this combining could be executed gradually for better accuracies.

VI. Conclusion

The best object recognition rates were achieved in the experiments by initially separating and subsequently combining convolution layers according to the positions of the tactile sensor patches, especially when making patch, finger, and hand maps (Pattern IV). This indicates that combining layers gradually, following the robots configuration, could be effective for CNNs. The proposed CNN could recognize a grasping object with one time step of tactile and joint information. Therefore, the CNN could also be applied to in-hand manipulation and grasp stability tasks in which quick processing is required. Nowadays, some of the differently shaped robots have distributed tactile sensors on their surface, for example, spider- or snake-shaped disaster robots. This proposal could be a suggestion for how to process the tactile information with CNNs for such robots.

In the future, we intend to work on the Graph Convolutional Network [32] inspired by the results of these proposed CNNs with morphology-related convolution.

REFERENCES

[1] Y. Chebotar, K. Hausman, Z. Su, G. S. Sukhatme, and S. Schaal, "Self-supervised regrasping using spatio-temporal tactile features and reinforcement learning," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2016, pp. 1960–1966.

[2] K.-T. Yu and A. Rodriguez, "Realtime state estimation with tactile and visual sensing for inserting a suction-held object," *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1628–1635, 2018.

[3] A. Montano and R. Surez, "Manipulation of unknown objects to improve the grasp quality using tactile information," vol. 18, p. 1412, 05 2018.

[4] Y. Gao, L. A. Hendricks, K. J. Kuchenbecker, and T. Darrell, "Deep Learning for Tactile Understanding From Visual and Haptic Data," *ArXiv e-prints*, Nov. 2015.

[5] M. Li, H. Yin, K. Tahara, and A. Billard, "Learning object-level impedance control for robust grasping and dexterous manipulation," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 6784–6791.

[6] A. Schmitz, Y. Bansho, K. Noda, H. Iwata, T. Ogata, and S. Sugano, "Tactile object recognition using deep learning and dropout," in *2014 IEEE-RAS International Conference on Humanoid Robots*, Nov 2014, pp. 1044–1050.

[7] S. Funabashi, A. Schmitz, T. Sato, S. Somlor, and S. Sugano, "Robust in-hand manipulation of variously sized and shaped objects," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept 2015, pp. 257–263.

[8] T. P. Tomo, W. K. Wong, A. Schmitz, H. Kristanto, A. Sarazin, L. Jamone, S. Somlor, and S. Sugano, "A modular, distributed, soft, 3-axis sensor system for robot hands," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, Nov 2016, pp. 454–460.

[9] T. P. Tomo, A. Schmitz, W. K. Wong, H. Kristanto, S. Somlor, J. Hwang, L. Jamone, and S. Sugano, "Covering a robot fingertip with uskin: A soft electronic skin with distributed 3-axis force sensitive elements for robot hands," *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 124–131, Jan 2018.

[10] S. Funabashi, S. Morikuni, A. Geier, A. Schmitz, S. Ogasa, T. P. Tomo, S. Somlor, and S. Sugano, "Object recognition through active sensing using a multi-fingered robot hand with 3d tactile sensors," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep 2018 (Accepted).

[11] A. J. Spiers, M. V. Liarokapis, B. Calli, and A. M. Dollar, "Single-grasp object classification and feature extraction with simple robot hands and tactile sensors," *IEEE Transactions on Haptics*, vol. 9, no. 2, pp. 207–220, April 2016.

[12] D. Feng, M. Kaboli, and G. Cheng, "Active prior tactile knowledge transfer for learning tactual properties of new objects," *Sensors*, vol. 18, no. 2, 2018.

[13] M. Kaboli, A. D. L. R. T, R. Walker, and G. Cheng, "In-hand object recognition via texture properties with robotic hands, artificial skin, and novel tactile descriptors," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, Nov 2015, pp. 1155–1160.

[14] T. Matsubara and K. Shibata, "Active tactile exploration with uncertainty and travel cost for fast shape estimation of unknown objects," *Robotics and Autonomous Systems*, vol. 91, pp. 314 – 326, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S092188901630522X

[15] P. Falco, S. Lu, A. Cirillo, C. Natale, S. Pirozzi, and D. Lee, "Cross-modal visuo-tactile object recognition using robotic active exploration," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 5273–5280.

[16] A. Vsquez, Z. Kappassov, and V. Perdereau, "In-hand object shape identification using invariant proprioceptive signatures," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2016, pp. 965–970.

[17] D. Cockbum, J. Roberge, T. Le, A. Maslyczyk, and V. Duchaine, "Grasp stability assessment through unsupervised feature learning of tactile images," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 2238–2244.

[18] H. Liu, D. Guo, and F. Sun, "Object recognition using tactile measurements: Kernel sparse coding methods," *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 3, pp. 656–665, March 2016.

[19] V. Kumar, A. Gupta, E. Todorov, and S. Levine, "Learning dexterous manipulation policies from experience and imitation," *CoRR*, vol. abs/1611.05095, 2016.

[20] M. Meier, F. Patzelt, R. Haschke, and H. J. Ritter, "Tactile convolutional networks for online slip and rotation detection," in *ICANN*, 2016.

[21] W. Yuan, C. Zhu, A. Owens, M. A. Srinivasan, and E. H. Adelson, "Shape-independent hardness estimation using deep learning and a gelsight tactile sensor," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 951–958.

[22] C. Larson, J. Spjut, R. A. Knepper, and R. F. Shepherd, "Orbtouch: Recognizing human touch in deformable interfaces with deep neural networks," *CoRR*, vol. abs/1706.02542, 2017.

[23] L. le Cao, K. Ramamohanarao, F. Sun, H. Li, W. bing Huang, and Z. M. M. Aye, "Efficient spatio-temporal tactile object recognition with randomized tiling convolutional networks in a hierarchical fusion strategy," in *AAAI*, 2016.

[24] M. Meier, G. Walck, R. Haschke, and H. J. Ritter, "Distinguishing sliding from slipping during object pushing," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2016, pp. 5579–5584.

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

[26] W. Yuan, Y. Mo, S. Wang, and E. Adelson, "Active Clothing Material Perception using Tactile Sensing and Deep Learning," *ArXiv e-prints*, Nov. 2017.

[27] R. Calandra, A. Owens, D. Jayaraman, J. Lin, W. Yuan, J. Malik, E. H. Adelson, and S. Levine, "More Than a Feeling: Learning to Grasp and Regrasp using Vision and Touch," *ArXiv e-prints*, May 2018.

[28] H. Iwata, R. Hayashi, Y. Shiozawa, and S. Sugano, "Basic control techniques of twendy-one hand which has passive flexibility - achievement of diverse grip and manipulation by transitions between grip forms," *26th Conference of Robotics Society of Japan*, 2008.

[29] M. Johannes, J. D. Bigelow, J. M. Burck, S. D. Harshbarger, M. Kozlowski, and T. Van Doren, "An overview of the developmental process for the modular prosthetic limb," vol. 30, pp. 207–216, 01 2011.

[30] J. Kwiatkowski, D. Cockburn, and V. Duchaine, "Grasp stability assessment through the fusion of proprioception and tactile signals using convolutional neural networks," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept 2017, pp. 286–292.

[31] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The ycb object and model set: Towards common benchmarks for manipulation research," in *2015 International Conference on Advanced Robotics (ICAR)*, July 2015, pp. 510–517.

[32] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, ser. NIPS'16. USA: Curran Associates Inc., 2016, pp. 3844–3852. [Online]. Available: http://dl.acm.org/citation.cfm?id=3157382.3157527