Fast Texture Classification Using Tactile Neural Coding and Spiking Neural Network

Tasbolat Taunyazov[1,2], Yansong Chua[3], Ruihan Gao[1,4], Harold Soh[2], Yan Wu[1]

Abstract— **Touch is arguably the most important sensing modality in physical interactions. However, tactile sensing has been largely under-explored in robotics applications owing to the complexity in making perceptual inferences until the recent advancements in machine learning or deep learning in particular. Touch perception is strongly influenced by both its temporal dimension similar to audition and its spatial dimension similar to vision. While spatial cues can be learned episodically, temporal cues compete against the system's response/reaction time to provide accurate inferences. In this paper, we propose a fast tactile-based texture classification framework which makes use of the spiking neural network to learn from the neural coding of the conventional tactile sensor readings. The framework is implemented and tested on two independent tactile datasets collected in sliding motion on 20 material textures. Our results show that the framework is able to make much more accurate inferences ahead of time as compared to that by the state-of-the-art learning approaches.**

I. INTRODUCTION

The sense of touch allows humans to make perceptual judgement on the environment that the skin comes into contact with, make timely decisions to correct a taken action and even infer social cues from interactions with others. Although it forms an indispensable part of our senses, we can hardly find tactile sensors on most social and service robots – the biomimetic agents to increasingly assist in our daily living. Unlike other senses, the sense of touch is a high-dimensional distributed system across the skin which gathers both spatial and temporal sensory cues. This nature makes it complex to be implemented and modelled in artificial systems. With the recent advances in material science and machine learning, research on tactile sensor-based perception and control starts to pick up momentum, including material classification [1], [2], object and shape recognition [3], [4], grasping [5], [6], slip detection [7] and manipulation [8], [4]. There is also a growing body of work on multimodal perception using both tactile and visual sensing modalities [9], [10] to improve inference performance.

*This research is partially supported by the Agency for Science, Technology and Research (ASTAR) under its AME Programmatic Funding Scheme (Project #A18A2b0046) and National Robotics Programme of Singapore (RECT Grant #1722500063).

[1]Robotics & Autonomous Systems Department, A*STAR Institute for Infocomm Research, Singapore. {gao_ruihan,wuy}@i2r.a-star.edu.sg

[2]School of Computing, National University of Singapore, Singapore. {tasbolat, hsoh}@comp.nus.edu.sg

[3]Central Research Institute, Huawei Technologies Co., Ltd, Shenzhen, China. chua.yam.song@huawei.com

[4]School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore.

Because of the importance of the temporal dimension in the sense of touch, many state-of-the-art frameworks on tactile perception need to acquire long temporal sequences of the tactile readings for both model training and inference [11], [2], [12]. This process is usually both power and data hungry. For a real-life system which depends on the sensory feedback for control-loop closure, such implementations may not be ideal. For example, when a robot comes into physical interaction with another agent or the environment, fast discrimination of material textures may be essential to correct any potentially unintended or dangerous actions, e.g. accidentally slapping a person, bumping into a wall versus a rope. Human, a power-efficient biological system on the other hand, seems to be capable of making fast tactile perceptual inferences.

In this paper, we propose a power-efficient tactile texture classification framework which makes use of the spiking neural network (SNN) to learn from the neural coding of the conventional tactile sensor measurements. This framework draws inspiration from the fact that the biological system largely codes and transmits signals through neurotransmitters, a.k.a spikes between the sensory receptors, neurons in the nervous system, and target muscle cells [13]. To validate the performance of this framework, independent tactile datasets collected from two unique tactile sensors with non-event-based outputs (the iCub RoboSkin and the SynTouch BioTac) sliding on 20 material textures are used for our experiments. The noise profile of the sensors and the data collection constraints are different for the two datasets, which allow us to scrutinise the different aspects of the framework. In particular, the BioTac dataset has higher temporal resolution and less sensor noise. It is attached onto a highly precise and accurate industrial collaborative robot – the KUKA LBR iiwa – to carry out the data collection in strict conditions of movement velocity and contact force. On the other hand, the RoboSkin dataset has lower temporal resolution and is collected in a more natural approach [2]. We hypothesise that for the cleaner BioTac dataset, given a suitable neural coding to encode the data into spike train, a simple linear classifier is sufficient to make inferences at early stage while SNNs can be used as a general approach for both scenarios for better inference performance.

In summary, the main contributions of this paper are:

- An overall texture classification framework that is computationally efficient and able to make fast inference. It can also be run on a neuromorphic chip such as the Intel Loihi;
- A neural coding scheme to convert raw tactile data from

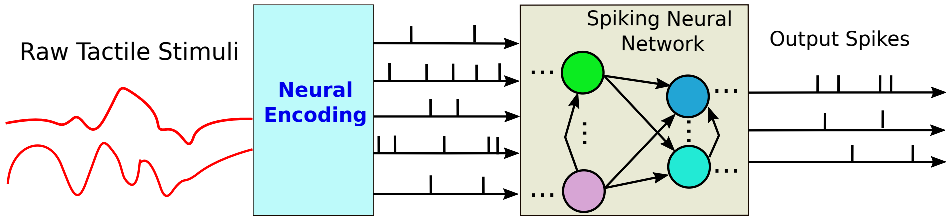


Fig. 1: An overview of texture classification using tactile data: raw tactile data of surface is encoded into spike trains. These spike trains are then learned to classify the material texture.

non-event-based sensors into sparse spike trains which encode important temporal information of the tactile signal;

- An SNN-based architecture tested on two tactile datasets to classify textures with high accuracy.

The rest of this paper is organized as follows: In Section II, related works on tactile-based perception are reviewed. The proposed framework is described in details in Section III. Section IV describes the experimental setup while the results and discussions are presented in Section V. Section VI concludes our research findings and proposes future work.

II. BACKGROUND AND RELATED WORK

Tactile perception is becoming a very active research area which includes tactile-based object classification [14], object manipulation [15], grasp stability prediction [16] and texture identification [10].

In this work, we focus on texture classification. In particular, we explore how spike trains can be generated from tactile data and how those spikes learnt to make decision. One prior work by [17] uses Izhikevich neuron model [18] to generate spike trains, particularly regular spiking neurons. It extracts coefficient of variation for interspike interval (ISI) and trains these features using k-NN algorithm. An accuracy of 78 % was achieves for 10 textures.

Friedl et al. [19] simulate two fast adapting and one slow adapting human mechanoreceptors in our skin. These are Pacianian cells (PC, operate at 30 - 700 Hz), Meisser cells (RA, operate at 1-60 Hz) and slow adapting Merkel cells (SA1). Merkel cells are mostly responsible for identifying static pressure. The spike trains for these neurons are generated using adaptive leaky-integrate-and-fire (aLIF) [20] neurons. These spike trains are learnt using one layer leaky-integrate-and-fire (LIF) neuron before frequency features are extracted from the output of this layer. These features are then trained using linear support vector machine (SVM). The model achieves 65.6 % accuracy in classifying 18 different materials.

Yi et al. [21] utilize the Izhikevich neuron model to generate spike trains. The Izhikevich model is defined by several parameters. Particularly, the parameters control the sensitivity of spikes and decay rate and also regulate the reset and adaptation times. The most typical Izhikevich neurons, namely the regular spiking neurons [17] are used. They collect different features from spike trains such as the Victor-Purpura Distance [22], the van Rossum Distance [23], ISI and the first spike latency. These features are then trained using a k-NN model which achieves $\approx$ 77.6 % ($k = 11$) accuracy for 8 different textures.

The latest work, to the best of our knowledge, for texture identification using spike trains is done by [24]. They generate spike trains, similar to [17], [21] using Izhikevich neuron model. However, unlike prior arts, this work generates three different spike trians: Slow Adapting (SA), based on regular spiking, FA-Rising and FA-Falling, both based on fast spiking neurons. Then these spike trains are trained using the Extreme Learning Machine (ELM), one layer SNN. This work is implemented in a neuromorphic chip and achieved an accuracy of 92 % for 10 objects.

The aforementioned works for texture classification have encountered some challenges. Firstly, it is difficult to tune Izhikevich neuron model. Its performance is very sensitive to its parameters while the parameter space is large. Secondly, using hand-crafted features may be task-specific and inefficient, since it is difficult to measure similarity between two spike trains. Thirdly, ELM may not be an appropriate method for applications in complex scenario, e.g. a significant increase in the number of textures.

Our work in this paper is a first attempt to address these challenges. We simplify the encoding scheme, while keeping the ability to represent more complex temporal information. We also propose a multi-layer training approach for texture identification using state-of-the-art SNN training framework.

III. METHODOLOGY

This section describes the proposed method to classify textures using non-event-based tactile data. Firstly, raw continuous temporal tactile data are encoded into spike trains. Here, we refer to spike trains as event-based temporal data in which it can be either 0 (non-spiking state) or 1 (spiking state) at any instance. Then, a texture classifier is trained using a Spiking Neural Network (SNN). The overall architecture for our approach is depicted in Fig. 1. Implementations using Artificial Neural Network (ANN) and SVM are used for benchmarking against SNNs.

A. Neural Encoding

We explore the use of simple thresholding technique to encode the raw tactile data into spike trains [25]. It uses K thresholds to generate spikes from a given time series data. Given raw tactile signal for i-th taxel, $y_i(t)$, the encoded

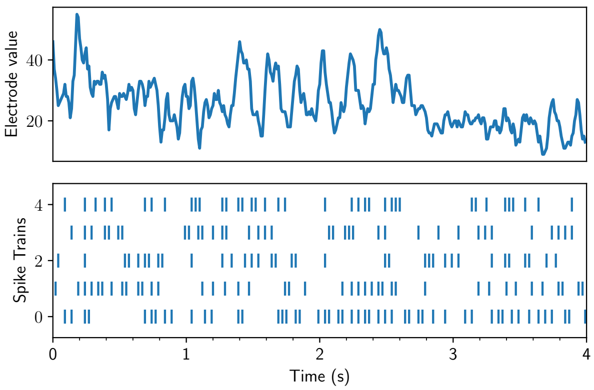


Fig. 2: An example of threshold encoding on tactile data when $K = 2$.

spike train for k-th neuron, $s_i^k(t)$, is defined as:

$$s_i^k(t) = \begin{cases} 1, & k \le K, y_i(t) \ge k, y_i(t-1) < k \\ 1, & k+1 \le k \le K, y_i(t) \le k/2, y_i(t) > k/2 \\ 1, & k = 2K+1, y_i(t) = \max\{y_i(t)\} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

where $k \in [1, ..., K, ..., 2K+1]$. In total, it outputs $2K+1$ spike trains. Generally, K is to be kept as small as possible as it scales the dimension for temporal data. We empirically chose $K = 2$ for both BioTac and RoboSkin data. An example of our neural encoding technique is given in Fig. 2.

B. Spike Response Model

We use the Spike Response Model (SRM) [26], [27] as our neural model for the training process. It is a generalization of the leaky integrate-and-fire model. Spikes are generated when membrane potential $u(t)$ exceeds a predefined threshold φ. The neuron in SRM depends on the incoming spikes to be convolved by a response kernel, $\epsilon(\cdot)$, and a refractory response, $\nu(\cdot)$:

$$u(t) = \sum w_i(\epsilon * s_i)(t) + (\nu * o)(t) \tag{2}$$

where w_i is a synaptic weight, $*$ indicates convolution, $s_i(t)$ are the incoming spikes from input i, and $o(t)$ is the neuron's output spike train.

C. Spike Data Training

The biggest challenge in training with event-based data is that the derivative of the spike generation function is undefined. There are possible ways to overcome this. For this work, we use Spike Layer Error Reassignment in Time (SLAYER) [27]. SLAYER uses a temporal credit assignment policy to back-propagate errors to preceding layers for update. SLAYER is simulated and trained in GPU hardware in offline manner. However, once trained in SLAYER, the SNN model can also be transferred to the neuromorphic hardware for efficient execution.

We use a simple 2-layer fully connected (FC) architecture for our models. The hidden layer size is set to 60 and 1200 for BioTac and RoboSkin respectively. Our neural encoding with $K = 2$ icreases raw tactile data size by five times. Hence, the input size for BioTac spike train is $5 \times 19 = 95$ where 19 is number of BioTac electrodes. The input size for RoboSkin is $5 \times 60 = 300$ where 60 is number of taxels on the RoboSkin of the iCub's forearm. Total sequence length for BioTac and RoboSkin tactile data is 400 ($4s$ sampled at 100 Hz) and 75 ($1.5s$ sampled at 50 Hz) respectively.

The input to the models is encoded spike train and output is also spike train with the size of number of classes. Thus, each output spike train corresponds to one specific class. We make decision for classification by looking at the highest number of spikes in output layer. The model is trained using the following loss function:

$$\mathcal{L} = \frac{1}{2}\sum_{t=0}^{T}\left(\sum \mathbf{s}^o(t) - \sum \tilde{\mathbf{s}}^o(t)\right)^2 \tag{3}$$

where $\sum \mathbf{s}^o(t)$ is a number of spike trains in the output layer and $\sum \tilde{\mathbf{s}}^o(t)$ is a desired spike counts.

SLAYER requires to define the desired output counts. For BioTac data, we set positive and negative spike counts to 250 and 30 respectively. For RoboSkin data, 70 and 3 are used respectively. These values are chosen empirically.

D. Comparison Models

ANN models: The ANN architecture consists of Long-Short-Term-Memory (LSTM) units [28]. LSTM is well-known for ability to learn temporal sequences [29]. It consists of cell state and hidden state, which are updated depending on the temporal pattern of the input data. The input to the models is raw tactile signal. The input sizes are 19 and 60 for BioTac and RoboSkin data respectively. In both cases, we keep hidden size as 60 and number of layers as 2. For RoboSkin data, we use Convectional Neural Network (CNN) to get high level features and supply them to LSTM. We apply softmax to the last value of the output sequence of LSTM to obtain the probability of belonging to a specific class.

SVM models: In order to assess the encoded spike trains on ability to represent the tactile signal, we use SVM with linear kernel on two different inputs. Firstly, we collapse raw tactile data in time dimension and use it as input features for SVM. Secondly, we perform the same operation on the encoded spike trains and use the resulting spike counts as inputs. While the input is a temporal sequence, we hypothesize that texture information is rather homogeneous in time. Rate coding [30], whereby input is coded as mean firing rates of neurons, may suffice in separating the classes. As such, spike counts may already achieve good classification accuracy for our data. Therefore, a simple SVM is used to investigate in both datasets.

We split the tactile data into training and testing sets with ratio of 7:3. Each model, except ANN models, are trained 5 times. Due to the computational cost, ANN models are trained only twice. We then report the mean accuracy results.

SNN models are trained for 300 epochs while ANNs are trained for 2000 epochs. ANNs are trained for longer because of the much higher number of trainable parameters involved.

IV. Experimental Setup

In this section, we describe the sensors and robot setup used for data collection.

A. iCub RoboSkin Tactile Sensor

The iCub is a tendon-driven humanoid robot with RoboSkin, capacitive tactile sensors on its skin [31]. The capacitive tactile sensor works on the principle of dielectric deformation [32]. The RoboSkin is distributed among hand, forearm, upper arm and torso with 18 patches. Each patch is made of triangular modules; and each module has 10 taxels. In our experiment, we use taxels from 6 patches from the forearm. In most cases, not all taxels are activated [2].

B. SynTouch BioTac Sensor

The SynTouch BioTac [33] is a multimodal tactile sensor that mimics touch sensory ability of a human finger. The sensor is filled with liquid. An externally exerted force is transmitted to the internal acoustic pressure sensor through this liquid medium. The BioTac sensor can sense pressure, vibrations and temperature simultaneously. In our experiment, we only use its pressure modality which comes from 19 electrodes.

C. iCub Data Collection

This dataset has been made publicly accessible with detailed data collection procedure given in [2]. Its setup has two major differences from other setups given in the literature for tactile texture identification:

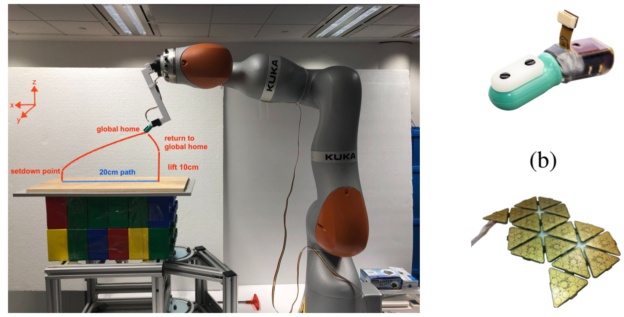
1) Due to the natural passive compliant nature of tendon-based actuation, the robot motion is not strictly under constant speed control.
2) No force control is exerted to the motion.

Originally, the RoboSkin dataset contains samples for 23 materials. However, to have better comparison with the BioTac dataset which is collected in [34], we only use the common 20 materials for this work. Each material consists of 50 samples. The RoboSkin is depicted in Fig. 3c.

D. BioTac Data Collection

The setup is very similar to [35]. However, instead of using a simple linear actuator, a KUKA LBR iiwa robot is used for data collection [34]. The iiwa robot is a 7-DoF robot known to be very precise and robust in task performance. The BioTac sensor (illustrated in Fig. 3b) is attached to the end-effector of iiwa robot as shown in Fig. 3a. The sliding movements are subject to constant force and velocity control as shown Fig. 4.

Force Control The change in the pressure value of BioTac sensor is linear to the magnitude of the contact normal force within range of 0-2 N as given in [35]. For instance, change in pressure value 40 corresponds to 0.4 N. Slight fluctuations on pressure value during sliding are expected.



(a) (b) (c)

Fig. 3: a) The KUKA LBR iiwa robot setup with BioTac Sensor [34]; b) BioTac sensor; and c) RoboSkin sensors.

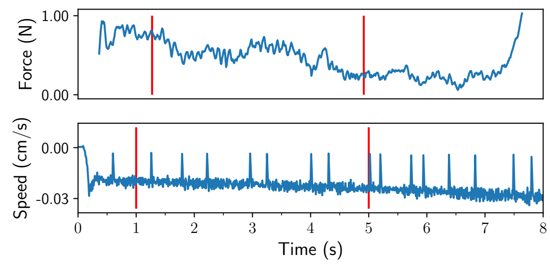


Fig. 4: An illustration of the force and speed profile of the BioTac sliding motion.

Velocity Control The KUKA LBR iiwa's Sunrise package [36] implemented with constant velocity control is used. It works on the principle of closed-loop control and receives the velocity feedback using Jacobian matrix. An illustration of the velocity profile is shown in Fig. 4.

While the exact data collection procedure can be found in [34], it is worth noting that the BioTac sensor moves over a linear trajectory for 20 cm with a constant speed of 2.5 cm/s, which is the optimal speed for texture classification given in [35]. With this setup, 20 different materials are collected with 50 samples each. The dataset has also been made public available by [34].

Due to the noise during the sliding motion, we only use data points between the 1^{st} and 5^{th} seconds.

V. Results and Discussions

A. Classification Results

Classification results for both BioTac and RoboSkin datasets are summarized in Table I. The accuracy scores for the BioTac dataset are similar for all models. It is approximately $\approx 94\%$. This is expected as the data collection

TABLE I: Texture Classification Accuracy Scores

Models	BioTac	iCub
SNN	0.946 (0.013)	0.922 (0.005)
ANN	0.945 (0.015)	0.935 (0.005)
SVM (spike)	0.935 (0.015)	0.633 (0.018)
SVM	0.942 (0.007)	0.505 (0.056)

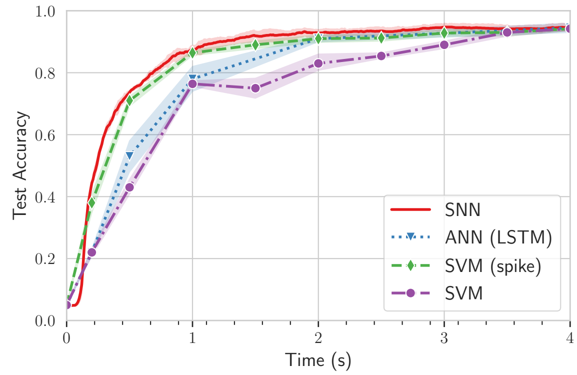


Fig. 5: Accuracy score over time for BioTac data using different models.

setup for BioTac is much stricter than that for RoboSkin, thus presenting a very clean dataset for easier inference.

The ANN for the RoboSkin dataset performs marginally better than SNN ($\approx$ 1 % difference). However, the SVM models are far more inferior to SNN and ANN. The classification reports for full spike counts (4 s - BioTac, 1.5 s - iCub). This suggests that iCubs data is more complex and/or noisy.

B. Fast Classification

One of the main advantages of SNN is to be able to make decision fast, at the early stages of output spike trains. This can be easily seen from output spike trains shown in Fig. 6. The correct class (indicated with red color) spikes more frequently than the rest.

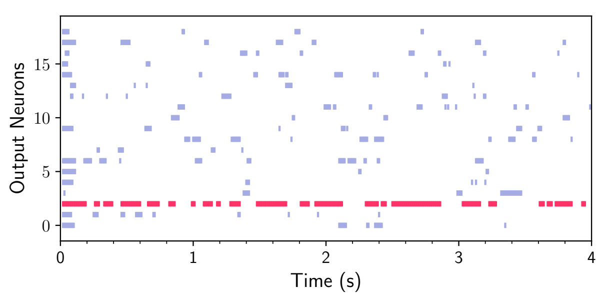


Fig. 6: Spike trains in output layer. Red indicates correct class.

Fig. 5 shows that SNN outperforms all other models in predicting correct class at early stage for BioTac tactile data. SVM on encoded spike counts performs very similarly to the SNN. This suggests that the encoded spike count already contains sufficient information to correctly classify. This helps to confirm our hypothesis that clean tactile data is primarily homogeneous in time. Hence spike counts based on how data is distributed across different threshold values alone gives comparable classification results with fully trained SNN. Surprisingly, SVM on raw data also gives comparably good result, further indicating the homogeneity of the input data even without being encoded to spike train. We note the accuracy dropping in SVM for raw tactile data at time 1.5 s. This maybe due to the spike in raw tactile data. To sum up, the threshold encoding helps to improve the linear separability of the data, and is useful in both SVM and SNN cases. Fig. 7 shows that t-SNE [37] on encoded data already divides the classes in visually separable clusters.

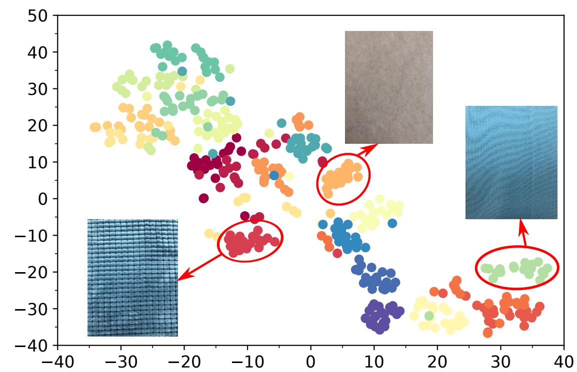


Fig. 7: t-SNE on encoded spike trains. We use van Rossum Distance [23] to calculate similarity between spike trains.

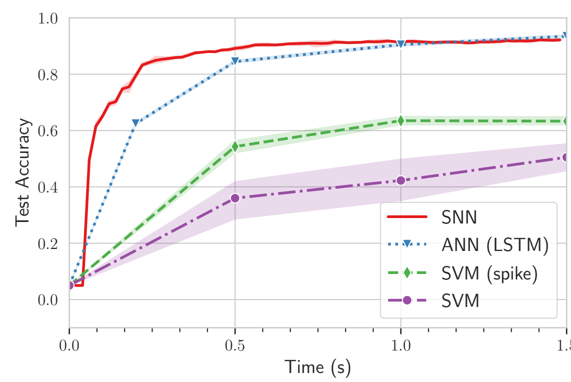


Fig. 8: Accuracy score over time for iCub data using different models.

Similarly, SNN outperforms other models for the RoboSkin tactile data to classify textures at early stages. However, given the full sequence, the ANN outperforms the SNN model. ANN and SNN models have similar result at $\approx$ 1 s. At the early stages, classification accuracy for ANN is much higher than the SVM models. This suggests that the data is more complex in nature and linear classifier on threshold encoding is insufficient to describe the data. The complexity of the data may not come from the observed (tactile surface) but from the observer (robot). Note that both SVM and ANN are trained for each time instance indicated with markers in Fig. 5 and 8.

It is also worth noting that for the non-spiking models, hand-crafted bin size in the temporal domain need to be empirically defined. While the smaller the bin size, the faster the robot can make a decision and a follow-up action. However, this bin size is task-specific and less practical for deployed system. However, SNN models do not pose such concerns as it makes inferences as each sample instance arrives.

VI. Conclusion and Future Work

In this work, we proposed an efficient texture classification framework for conventional tactile-sensors. We used thresholding encoding technique to generate spike trains for two sets of tactile sensor data. In the less set of noisy data, the spike trains are homogeneous in time and can be easily classified using a simple linear classifier. In a general scenario, the encoding can be used to train an SNN to improve the classification accuracy comparable to ANN but with fast inference. Future work includes the investigation on the representation of the spatial information in the model to better predict the tactile events. Our dataset and code for our models are available online at `https://dexrob.github.io/dexrob/fast_texture_classification_iros_2020/`.

References

[1] N. Jamali and C. Sammut, "Majority voting: Material classification by tactile sensing using surface texture," *IEEE Transactions on Robotics*, vol. 27, no. 3, pp. 508–521, 2011.

[2] T. Taunyazov, H. F. Koh, Y. Wu, C. Cai, and H. Soh, "Towards effective tactile identification of textures using a hybrid touch approach," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 4269–4275, IEEE, 2019.

[3] H. Soh and Y. Demiris, "Incrementally learning objects by touch: Online discriminative and generative models for tactile-based recognition," *IEEE transactions on haptics*, vol. 7, no. 4, pp. 512–525, 2014.

[4] J. Sanchez, C. M. Mateo, J. A. Corrales, B.-C. Bouzgarrou, and Y. Mezouar, "Online shape estimation based on tactile sensing and deformation modeling for robot manipulation," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 504–511, IEEE, 2018.

[5] Y. Bekiroglu, K. Huebner, and D. Kragic, "Integrating grasp planning with online stability assessment using tactile sensing," in *2011 IEEE International Conference on Robotics and Automation*, pp. 4750–4755, IEEE, 2011.

[6] Y. Su, Y. Wu, K. Lee, Z. Du, and Y. Demiris, "Robust grasping for an under-actuated anthropomorphic hand under object position uncertainty," in *Proceedings of the 12th IEEE-RAS International Conference on Humanoid Robots. Humanoid 2012.*, pp. 719–725, Nov 2012.

[7] T. Taunyazoz, W. Sng, H. H. See, B. Lim, J. Kuan, A. F. Ansari, B. Tee, and H. Soh, "Event-driven visual-tactile sensing and learning for robots," in *Proceedings of Robotics: Science and Systems*, July 2020.

[8] Y. Su, Y. Wu, H. Soh, Z. Du, and Y. Demiris, "Enhanced kinematic model for dexterous manipulation with an underactuated hand," in *Proceedings of 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS 2013.*, pp. 2493–2499, Nov 2013.

[9] F. Sun, C. Liu, W. Huang, and J. Zhang, "Object classification and grasp planning using visual and tactile sensing," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 7, pp. 969–979, 2016.

[10] S. Luo, W. Yuan, E. Adelson, A. G. Cohn, and R. Fuentes, "Vitac: Feature sharing between vision and tactile sensing for cloth texture recognition," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2722–2727, IEEE, 2018.

[11] 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)*, pp. 8–15, IEEE, 2016.

[12] D. Xu, G. E. Loeb, and J. A. Fishel, "Tactile identification of objects using bayesian exploration," in *2013 IEEE International Conference on Robotics and Automation*, pp. 3056–3061, IEEE, 2013.

[13] J. Daly, J. Brown, and J. Weng, "Neuromorphic motivated systems," in *The 2011 International Joint Conference on Neural Networks*, pp. 2917–2924, IEEE, 2011.

[14] H. Liu, Y. Yu, F. Sun, and J. Gu, "Visual–tactile fusion for object recognition," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 2, pp. 996–1008, 2016.

[15] 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," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3300–3307, 2018.

[16] J. Li, S. Dong, and E. Adelson, "Slip detection with combined tactile and visual information," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7772–7777, IEEE, 2018.

[17] U. B. Rongala, A. Mazzoni, and C. M. Oddo, "Neuromorphic artificial touch for categorization of naturalistic textures," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 4, pp. 819–829, 2015.

[18] E. M. Izhikevich, "Simple model of spiking neurons," *IEEE Transactions on neural networks*, vol. 14, no. 6, pp. 1569–1572, 2003.

[19] K. E. Friedl, A. R. Voelker, A. Peer, and C. Eliasmith, "Human-inspired neurorobotic system for classifying surface textures by touch," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 516–523, 2016.

[20] G. Indiveri, "A low-power adaptive integrate-and-fire neuron circuit," in *Proceedings of the 2003 International Symposium on Circuits and Systems, 2003. ISCAS'03.*, vol. 4, pp. IV–IV, IEEE, 2003.

[21] Y. Zhengkun and Z. Yilei, "Recognizing tactile surface roughness with a biomimetic fingertip: A soft neuromorphic approach," *Neurocomputing*, vol. 244, pp. 102–111, 2017.

[22] J. D. Victor and K. P. Purpura, "Nature and precision of temporal coding in visual cortex: a metric-space analysis," *Journal of neurophysiology*, vol. 76, no. 2, pp. 1310–1326, 1996.

[23] M. v. Rossum, "A novel spike distance," *Neural computation*, vol. 13, no. 4, pp. 751–763, 2001.

[24] M. Rasouli, Y. Chen, A. Basu, S. L. Kukreja, and N. V. Thakor, "An extreme learning machine-based neuromorphic tactile sensing system for texture recognition," *IEEE transactions on biomedical circuits and systems*, vol. 12, no. 2, pp. 313–325, 2018.

[25] Z. Pan, Y. Chua, J. Wu, M. Zhang, H. Li, and E. Ambikairajah, "An efficient and perceptually motivated auditory neural encoding and decoding algorithm for spiking neural networks," *Frontiers in Neuroscience*, vol. 13, p. 1420, 2020.

[26] W. Gerstner, "Time structure of the activity in neural network models," *Physical review E*, vol. 51, no. 1, p. 738, 1995.

[27] S. B. Shrestha and G. Orchard, "Slayer: Spike layer error reassignment in time," in *Advances in Neural Information Processing Systems*, pp. 1412–1421, 2018.

[28] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[29] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with lstm," 1999.

[30] W. Gerstner and W. M. Kistler, *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge university press, 2002.

[31] G. Metta, G. Sandini, D. Vernon, L. Natale, and F. Nori, "The icub humanoid robot: an open platform for research in embodied cognition," in *Proceedings of the 8th workshop on performance metrics for intelligent systems*, pp. 50–56, 2008.

[32] M. Fumagalli, S. Ivaldi, M. Randazzo, L. Natale, G. Metta, G. Sandini, and F. Nori, "Force feedback exploiting tactile and proximal force/torque sensing," *Autonomous Robots*, vol. 33, no. 4, pp. 381–398, 2012.

[33] J. A. Fishel and G. E. Loeb, "Sensing tactile microvibrations with the biotac—comparison with human sensitivity," in *2012 4th IEEE RAS & EMBS international conference on biomedical robotics and biomechatronics (BioRob)*, pp. 1122–1127, IEEE, 2012.

[34] R. Gao, T. Taunyazov, Z. Lin, and Y. Wu, "Supervised autoencoder joint learning on heterogeneous tactile sensory data: Improving material classification performance," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (Las Vegas, USA), IEEE, 2020.

[35] J. A. Fishel and G. E. Loeb, "Bayesian exploration for intelligent identification of textures," *Frontiers in neurorobotics*, vol. 6, p. 4, 2012.

[36] M. Safeea and P. Neto, "Kuka sunrise toolbox: Interfacing collaborative robots with matlab," *IEEE Robotics & Automation Magazine*, vol. 26, no. 1, pp. 91–96, 2018.

[37] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.