2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Visuo-Tactile Zero-Shot Object Recognition with Vision-Language Model

Shiori Ueda[1], Atsushi Hashimoto[2], Masashi Hamaya[2], Kazutoshi Tanaka[2], Hideo Saito[1]

***Abstract*— Tactile perception is vital, especially when distinguishing visually similar objects. We propose an approach to incorporate tactile data into a Vision-Language Model (VLM) for visuo-tactile zero-shot object recognition. Our approach leverages the zero-shot capability of VLMs to infer tactile properties from the names of tactilely similar objects. The proposed method translates tactile data into a textual description solely by annotating object names for each tactile sequence during training, making it adaptable to various contexts with low training costs. The proposed method was evaluated on the FoodReplica and Cube datasets, demonstrating its effectiveness in recognizing objects that are difficult to distinguish by vision alone.**

I. INTRODUCTION

Tactile perception is essential in recognizing and interacting with objects for both humans and robots equally [1]. While humans can visually estimate some of an object's properties like shape and material [2], we rely on touch to perceive other properties, such as hardness and elasticity. Vision alone can be misleading, particularly when distinguishing objects that appear similar or have been shaped uniformly. This issue extends to identifying the internal conditions of objects, like being boiled or frozen. Tactile data is often vital to complement visual data. On the other hand, its necessity is highly context-dependent. Hence, a method should be adaptable to each context with low training costs in visuo-tactile applications.

We focus on the task of zero-shot object recognition using both visual and tactile data. Zero-shot object recognition involves recognizing objects that are not included in the training dataset. This study considers leveraging the zero-shot ability of vision language models (VLMs) [3], [4]. VLM is one of the extensions of large language models (LLMs) that accept images and texts as inputs. Some studies report that LLMs acquire common sense [5], [6]. We expect that VLMs also possess common sense since they are developed based on LLMs. Such common sense would help estimate tactile properties from the similarity to a known object.

Tactile sensing involves a complex mix of actions and sensors. Zero-shot performance on various vision tasks has dramatically improved in the last few years [7]. This is due to the development of VLMs pre-trained on large-scale datasets [8]. Image and text data have standardized formats, allowing for unified data processing. On the other hand, each new action requires new data collection for tactile sensing. A greedy approach for incorporating tactile data into VLMs is fine-tuning VLMs for each new action or sensor. This method requires storing fine-tuned VLM models for each action and sensor, which would be intractable. Instead, we focus on bridging tactile information to a VLM through object names without fine-tuning them.

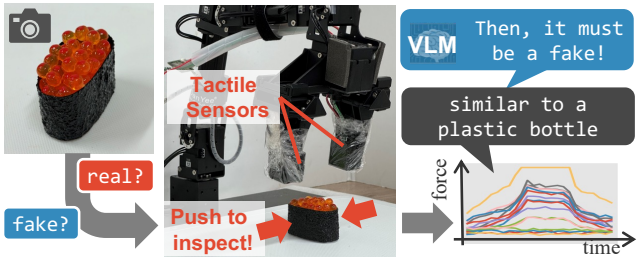


Fig. 1. Task overview. We are addressing the problem of recognizing objects that are difficult to distinguish by vision alone. To facilitate recognition, the robot collects tactile information in addition to visual observation. Tactile signals are converted into a text description (e.g., "similar to `${known reference object name(s)}`") and fed to VLM. Based on the image, the textual tactile description, and common sense, VLM identifies the object class in a zero-shot manner.

Based on these motivations, we propose a new approach for incorporating tactile data into the VLM for visuo-tactile zero-shot object recognition. Fig. 1 illustrates the task overview. The proposed method utilizes a VLM solely by retrieving tactilely similar objects, which requires only annotating object names for each sample. Moreover, tactile similarities are learned without images or semantic labels, enabling the method to be easily replaced whenever the action or sensor changes.

We developed two visuo-tactile datasets, the FoodReplica dataset and the Cube dataset, for evaluation. They simulate the challenging scenario of recognizing objects by vision alone (real foods and replicas, and cube-shaped foods with different internal conditions). The evaluation on these datasets demonstrates that the proposed method outperforms the visual-only method.

Our contributions are as follows:

- We propose a new approach incorporating tactile data into the VLM for visuo-tactile zero-shot object recognition. It leverages the common sense of VLMs to infer tactile properties from the names of tactilely similar objects.

This work was supported by JSPS KAKENHI Grant Number 21H04910, JST SPRING Grant Number JPMJSP2123 and Grant-in-Aid for JSPS Research Fellow Grant Number JP24KJ1962.

[1] S. Ueda and H. Saito are with Keio University, Yokohama 223-8522, Japan. shiori.ueda@keio.jp

[2] A. Hashimoto, M. Hamaya, and K. Tanaka are with OMRON SINIC X Corporation, Hongo 5-24-5, Bunkyo-ku, Tokyo, Japan atsushi.hashimoto@sinicx.com

- We introduce a tactile-to-text database that converts the tactile embedding into a textual description. It can be constructed with only tactile data and object names, facilitating adaptation to new actions or sensors at a low cost.
- We build two visuo-tactile datasets, FoodReplica and Cube, to simulate the challenging scenario of recognizing objects by vision alone.

II. RELATED WORKS

A. Visuo-tactile Applications

One of the common visuo-tactile applications is cross-modal learning [9], [10], [11], which aims at extracting shared features and mapping them between different modalities. Some studies have exploited this property of cross-modal learning and applied it to zero-shot recognition [12], [13], [14]. Liu *et al.* [12] employed dictionary learning to transfer knowledge from visual to tactile data and utilized semantic labels to calculate similarity. Fang *et al.* [13] introduced a conditional flow module to bidirectionally map the latent spaces of visual and tactile data. Yang *et al.* [14] aligned tactile embedding with a pre-trained image embedding to bind the tactile information to LLMs. These studies assume that the tactile sensation of an object can be estimated from its visual appearance. Our study differs from these studies in the sense that tactile data is used to obtain what cannot be estimated from visual data.

Visuo-tactile fusion is the other application, which aims to integrate different modalities to improve the performance of a task. Our study is grouped into this type of application. In previous studies in object recognition, combining visual and tactile data has been shown to improve the recognition accuracy [15], [16], [17]. Visuo-tactile fusion has also been applied to zero-shot recognition [18], [19]. Abderrahmane *et al.* [18] achieved zero-shot object recognition by predicting semantic labels from visual and tactile data. However, manually designed semantic labels are subjective and may not be consistent across different users. Experiments with multiple humans are needed to obtain a commonly recognized semantic label [20], [21], which is costly and limited to closed-set classes. Fu *et al.* [19] utilized the VLM to generate semantic labels from visual and tactile data. It successfully aligns visual data, tactile data, and semantic labels, although the format of tactile data is fixed, and fine-tuning the VLM is required. Instead, we propose to bridge tactile data to a VLM via names of objects similar to the recognition target in tactile. The VLM infers the target's tactile properties based on their similarity to known objects, which requires no fine-tuning. In addition, the similarity is calculated independently from other modalities. Thus, this approach essentially assumes no specific data format.

B. Actions for Tactile Perception

Grasping is often used for tactile object recognition because it enables the collection of rich tactile data needed to identify the shapes and materials of objects from various positions and angles [22], [23]. The actions can be much simpler if the goal is solely to identify tactile properties like hardness or material type. Yuan *et al.* [24] estimated the hardness of objects by pressing the target objects manually or by a robot hand in the normal direction. Several studies [25], [26], [27] recognized materials by sweeping sensors on the target objects. Yuan *et al.* [28] predicted the material of the clothing by gripping the wrinkle of the clothing with a parallel gripper. In this study, a pushing action is chosen rather than a grasping or sliding action to collect tactile data. It can be easily applied to various objects and can estimate hardness, which is difficult to infer by vision alone.

C. Tactile Recognition of Foods

Recognizing and handling fragile objects like foods is important for robotic applications [29]. Several studies focus on recognizing the tactile properties of foods [30], [31], but they often break the objects to measure these properties. Gemici *et al.* [30] presented a method for obtaining the tactile properties of foods by interacting with them using forks and knives. Sawhney *et al.* [31] constructed a dataset that includes the tactile properties of foods by squeezing them, pushing, and dropping their cut pieces. Yuan *et al.* [24] estimated the hardness of objects, including foods (tomatoes), by pressing them without breaking them, but they do not provide a way to classify the foods. In this work, we aim to recognize the tactile properties of objects by softly pushing them with a parallel gripper to avoid breaking them, including fragile objects like macarons.

III. PROPOSED METHOD

A. Problem Statement

Our task is zero-shot object recognition using visual and tactile data. Specifically, we aim to leverage tactile data as a cue to recognize objects that are difficult to distinguish using vision alone, as shown in Fig. 1.

Let $\boldsymbol{v}$ be the input RGB image, and let $\boldsymbol{X} = (\boldsymbol{x}_t)_{t=1}^{T}$ be the input tactile sequence, where $\boldsymbol{x}_t \in \mathbb{R}^{d_\mathrm{x}}$ is the tactile signal at time t, and T is the number of timesteps of the sequence. The output is a class label $y \in \mathcal{Y}_{\mathrm{unseen}}$, where $\mathcal{Y}_{\mathrm{unseen}}$ is the set of object classes given only at test time (and thus, inaccessible during the training phase).

B. Tactile Embedding Network

As illustrated in Fig. 2, we extract a tactile embedding $\boldsymbol{z} \in \mathbb{R}^{d_z}$ from the tactile sequence $\boldsymbol{X} \in \mathbb{R}^{T \times d_x}$ in an unsupervised manner. We employ a sequence-to-sequence (Seq2Seq) model [32] to learn the tactile embedding by reconstructing the input tactile sequence. Moreover, a vector quantization layer [33] is applied to convert the tactile embedding into a discrete latent variable. We construct a compact but discriminative model using a vector-quantization layer.

Let $Encoder$ be an RNN encoder, VQ be a vector quantization layer, and $Decoder$ be an RNN decoder. The tactile sequence $\boldsymbol{X}$ is first input to $Encoder$. $Encoder$ outputs $\boldsymbol{z}$, where $\boldsymbol{z}$ is the hidden state of the RNN cell at the last timestep. VQ quantizes the tactile embedding

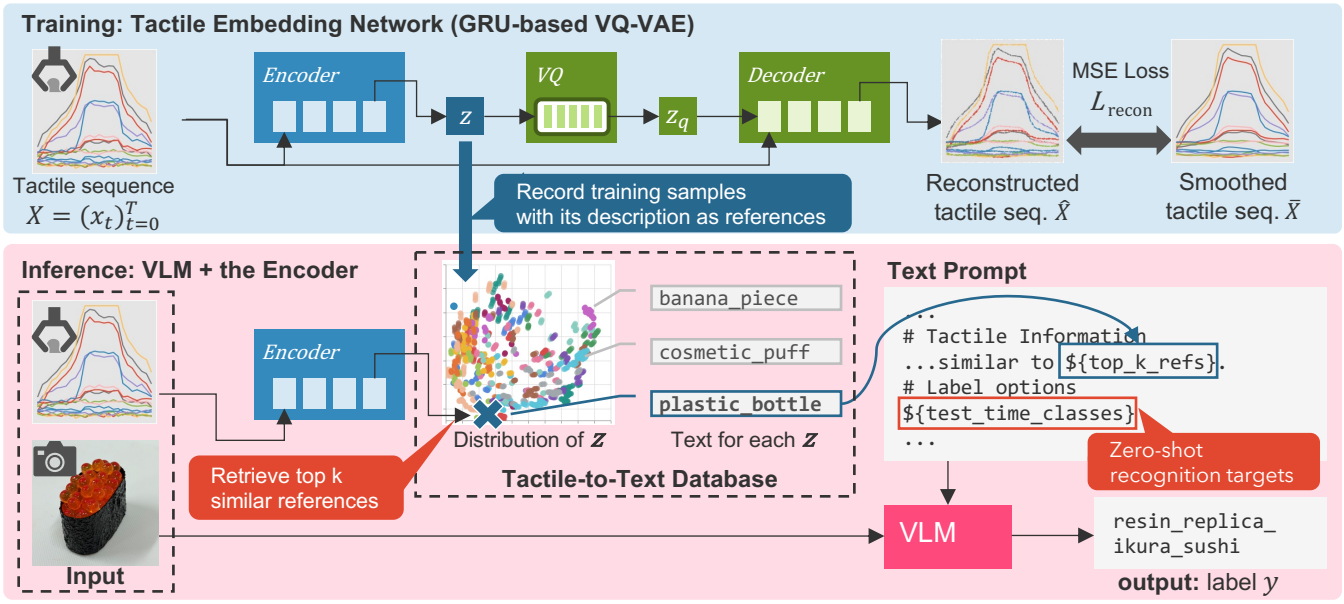


Fig. 2. The proposed pipeline. The tactile embedding network learns the tactile embedding from the tactile sequence. These embeddings are converted to textual descriptions in the tactile-to-text database. During inference, the Vision Language Model (VLM) receives a textual description along with the visual image and outputs the most likely class label for the input object in a zero-shot manner.

$\boldsymbol{z}$ into a discrete tactile embedding $\boldsymbol{z}_q$. In VQ, the input $\boldsymbol{z}$ is compared with a set of learnable codebook vectors $\mathcal{Z}_{\text{codebook}} \in \mathbb{R}^{K \times d_z}$, where K is the number of codebook vectors. The codebook vector $\boldsymbol{z} \in \mathcal{Z}_{\text{codebook}}$ that is closest to the input $\boldsymbol{z}$ is selected as the quantized tactile embedding $\boldsymbol{z}_q$. The output of VQ is then used as the initial hidden state of $Decoder$. With this initial hidden state $\boldsymbol{z}_q$ and $\boldsymbol{X}$, $Decoder$ decodes a sequence $\hat{X} = (\hat{\boldsymbol{x}}_t)_{t=1}^{T}$, which estimates a smoothed input $\bar{X} = (\bar{\boldsymbol{x}}_t)_{t=1}^{T}$. Here, we apply a smoothing operation to prevent the model from overfitting to the noise.

The loss function L follows VQ-VAE [33] and is defined as:

$$L = L_{\text{recon}} + L_{\text{vq}} + \beta L_{\text{commit}}, \tag{1}$$

where L_{recon} is the reconstruction loss, L_{vq} is the vector quantization loss, and L_{commit} is the commitment loss. L_{recon} is defined as the mean squared error (MSE) between $\bar{\boldsymbol{X}}$ and $\hat{\boldsymbol{X}}$. L_{vq} is the MSE between $\boldsymbol{z}$ and $\boldsymbol{z}_q$, where $\boldsymbol{z}$ is detached from the computational graph. L_{commit} is also the MSE between $\boldsymbol{z}$ and $\boldsymbol{z}_q$, but with detached $\boldsymbol{z}_q$. β is a hyperparameter that controls the strength of L_{commit}.

C. Tactile-to-Text Database

The tactile embedding $\boldsymbol{z}$ needs to be converted to a textual description to leverage the zero-shot ability of a VLM. To achieve this, we construct a tactile embedding database. The tactile-to-text database $D_{\text{tac2txt}} : \mathcal{Z}_{\text{ref}} \rightarrow \mathcal{Y}_{\text{ref}}$ is a dict-type database that contains pairs of tactile embeddings and class labels, where $\mathcal{Z}_{\text{ref}}$ is the set of tactile embeddings obtained from the TactileReference dataset. The labels of the dataset are tied to textual descriptions. To help the VLM better identify the tactile property of objects belonging to an unseen class $y \in \mathcal{Y}_{\text{unseen}}$, we refer to top-$k$ nearest classes from $\mathcal{Y}_{\text{ref}}$. Let z be a tactile embedding obtained from an unseen class object at inference, and let $z_i \in \mathcal{Z}_{\text{ref}}$ be the reference tactile embeddings. We retrieve the top-k nearest classes with the L2-norm distance $d(\boldsymbol{z}, \boldsymbol{z}_i) = \|\boldsymbol{z} - \boldsymbol{z}_i\|$. We then concatenate textual descriptions for the top-k classes with a comma to form $description$, which is included in the prompt for the VLM.

D. Vision-Language Model as a Zero-shot Classifer

We utilize a VLM to perform zero-shot object recognition using vision and tactile textual data. We use GPT-4V [4] as the VLM. The image $\boldsymbol{v}$ and the textual tactile representation $description$ obtained in the previous subsection are input to the VLM. The prompt is designed to compel the VLM to evaluate visual and tactile likelihoods together. Fig. 3 shows the exact prompt used in this study. The prompt is a template with two slots: `topk_refs` is replaced with $description$, and `test_time_classes` is replaced with $\mathcal{Y}_{\text{unseen}}$. The VLM selects a label y from $\mathcal{Y}_{\text{unseen}}$ based on $\boldsymbol{v}$ and $description$ with this prompt.

IV. EXPERIMENTAL SETUP

A. Datasets

We conducted experiments to evaluate the proposed method. We collected a tactile dataset, *TactileReference*, to train the tactile embedding network and construct the tactile-to-text database. In addition, we prepared two visuo-tactile datasets, *FoodReplica* and *Cube*, for testing.

Fig. 4 shows the TactileReference dataset. It consists of 32 classes of labels $y \in \mathcal{Y}_{\text{ref}}$ and the corresponding tactile sequences $\boldsymbol{X}$. For network training, the dataset was split into 27 classes for training and 5 classes for validation. The validation set was used to monitor the convergence

```
# Instruction
Identify the label of the object in the Image.
Follow the steps below to make your prediction:
1. Tactile Analysis: Assign a score (0-10) based
on how well the tactile information below matches
each label's expected property (e.g., stiffness).
Ignore the image at this stage.
2. Visual Analysis: Assign a score (0-10) based
on how well the image matches each label's
expected appearance. Ignore the tactile
information at this stage.
3. Label Selection: Sum the tactile and visual
scores for each label. The label with the highest
sum is chosen as the correct one.

# Tactile information
The mechanical stimulus (e.g., stiffness) when
touching it is similar to ${topk_refs}.
Note that there is no direct correlation between
the tactile reference provided and the categories
of the object.

# Label options
${test_time_classes}

# Output Format
```
label - tactile - visual - total
{label} - {tactile score} - {visual score} -
{total score} # Repeat for each option
Answer: {selected label}
```
```

Fig. 3. Prompt for visuo-tactile zero-shot object recognition. ${topk_refs} represents the reference classes of the top-k nearest tactile embeddings. ${test_time_classes} denotes the set of labels of the test dataset.

of the training and to determine the hyperparameters. We determined the text description corresponding to each label y after verifying that GPT had not misinterpreted the object. Note that we used only the training set to train the tactile embedding network but used both the training and validation sets for organizing the tactile-to-text database.

The FoodReplica dataset, shown in Fig. 5 (a), was used to evaluate the performance of the proposed method. A dataset sample is given as a triplet $(\boldsymbol{X}, \boldsymbol{v}, y)$. Food replicas are wax or plastic models of food. They are visually similar to real food but have different tactile properties. We chose food replicas as the target objects because they are among the most difficult objects to distinguish using vision alone. It consists of 22 classes (11 pairs of real foods and their replicas) of labels $y \in \mathcal{Y}_{\text{unseen}}$, visual images $\boldsymbol{v}$, and tactile sequences $\boldsymbol{X}$.

The Cube dataset, shown in Fig. 6 (a), was also used to evaluate the performance of the proposed method, with samples also given as $(\boldsymbol{X}, \boldsymbol{v}, y)$. It consists of 4 classes (2 objects $\times$ 2 conditions) of labels $y \in \mathcal{Y}_{\text{unseen}}$, visual images $\boldsymbol{v}$, and tactile sequences $(\boldsymbol{x}_t)$. It simulates a cooking situation, where objects are processed to have unified shapes but different internal conditions. We used kabocha_squash and kiri_mochi (a type of rice cake) as the target objects. We prepared two conditions for each object: one is a raw condition and the other is a boiled condition. Both objects become soft when boiled while preserving their visual appearance. We prepared this dataset to simulate a more practical context than FoodReplica.

B. Hardware Settings

The middle of Fig. 1 shows our hardware settings. We used a robot arm (OpenMANIPULATOR-X, RM-X52-TNM, Robotis Co., Ltd.) [34] with a parallel gripper. The distributed 3-axis tactile sensor (uSkin XR 1944, XELA Robotics Co., Ltd.) [35] was attached to each of the two fingers of the gripper. The tactile sensor has a four-by-four taxel array. With these settings, we can obtain $d_{\text{x}} = 4{\times}4{\times}3{\times}2 = 96$-dimensional tactile signals at each timestep.

For the image input of the test datasets, we used an iPhone 15 Pro rear camera (77 mm, f/2.8). During image capture, the target object was placed on a white cutting board for the FoodReplica dataset to simplify the background. For the Cube dataset, the target object was placed on an aluminum foil to ensure that the boundary between the white object and the background was visually distinguishable.

C. Data Collection

Tactile sequences were obtained by softly pushing the target object with the gripper following predefined motions. The motions were as follows:

1) Open the gripper above the target object.
2) Lower the gripper to the position where the bottom of it is almost touching the surface of the table.
3) Close the gripper until the norm of the tactile signal exceeds a threshold x_{th}.
4) Stop the gripper for $\Delta t_{\text{stopping}}$. Record the timestamp t_{stop} when the gripper stops. It is used for preprocessing the sequence.
5) Open the gripper and go back to the initial position.

We set $x_{\text{th}} = 0.004$ [1] as a sufficiently gentle value and $\Delta t_{\text{stopping}} = 0.2$ s as a time long enough to observe delayed physical reactions, we repeated the process ten times for each label y. The target object was placed below the gripper and slightly moved for each process to make the tactile sequence different.

An ideal sequence $\boldsymbol{X}$ starts when the gripper contacts the object and ends when the object is released. Unfortunately, it is difficult to detect such points precisely for versatile objects. Instead, we crop the sequence centered on the most vital contact: the timing of stopping the gripper, as shown in Fig. 7. t_{stop} is the timing when the gripper stops its motion with the strongest force. We recorded the sequence from 0.4 s before the t_{stop} (during the gripper closing motion) to 0.6 s after the t_{stop} (0.2 s for $\Delta t_{\text{stopping}}$ and 0.4 s during the gripper opening motion), which is a total of 1.0 s.

We implemented the above with a simple data augmentation by subsampling. We observed the tactile data with a

[1]The sensors were not calibrated and the value is sensor specific. We set $x_{\text{th}} = 0.003$ for gelatins exceptionally because the objects are far more fragile than others.

Fig. 4. Snapshots of the TactileReference dataset. It is used in two modules. In the tactile embedding network module, the dataset is split into 27 classes for training and 5 classes for validation. In the tactile-to-text database module, all classes of the dataset are used to construct the tactile-to-text database.

sampling rate of 125 Hz. Then, we subsampled the sequence at 25 Hz. By shifting the offset, we can obtain 5 different sequences from one process in this setting. Overall, based on the timing of t_{stop}, we cropped the sequence with $T = 25$ with 5 augmented samples.

In addition to the above subsampling, we obtained $\bar{\boldsymbol{X}}$ by applying a Gaussian filter to the observed sequence with its standard deviation of $\sigma = 3.0$. To summarize, the number of samples for each y was 50 (10 processes $\times$ 5 augmentations). This resulted in a total of 1350 samples (27 classes) for training and 250 (5 classes) for validation sets of TactileReference. For the other two test datasets (Food Replica and Cube), we collected 1 sample for each process, resulting in 10 samples from 10 processes for each $y \in \mathcal{Y}_{\text{unseen}}$.

D. Parameters for each module

For *Encoder*, we used a 2-layer bi-directional GRU with a hidden size of 32, followed by a linear layer with an output size of 4. For *Decoder*, we used a 2-layer GRU with a hidden size of 32. For *VQ*, we set d_{z} to 4 and the number of codebook vectors K to 32. The output of *VQ* was input to a linear layer with an output size of 32, which was then set as the initial hidden state of the decoder *Decoder*. In the loss function, we set β to 0.25. We implemented our network using PyTorch 2.0.0 and trained it on a single NVIDIA GeForce RTX 3080. We used the Adam optimizer with a learning rate of 0.01. The batch size was 256. We trained the network until the validation loss converged (22 epochs).

During inference, we set $k = 3$ and used the top 3 class labels to construct the tactile text *description*. We set the VLM's temperature to 0.0 to make the output deterministic.

E. Evaluation Metrics

We used accuracy as the evaluation metric, which is defined as the ratio of correctly recognized samples to the total number of test samples. We also used the confusion matrix to visualize the failure tendencies.

V. RESULTS

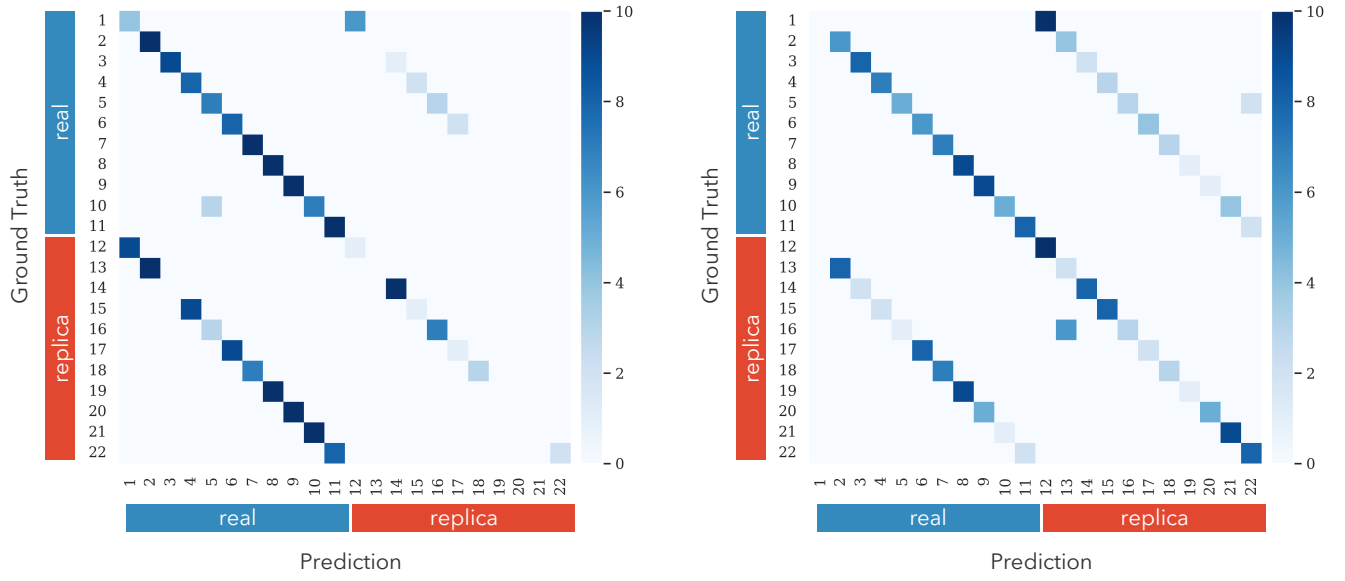
We evaluated the proposed method on the FoodReplica and Cube datasets. We also provide two ablation studies, one with a different choice of VLMs and one with a different size of the TactileReference dataset.

A. Comparison with the Vision-only Baseline

We compared the proposed method with the baseline, vision-only recognition using GPT-4V. It takes the image $\boldsymbol{v}$ as input. For the baseline, we removed instructions related to tactile data from the proposed prompt, as shown in Fig. 8. In Tab. I, the first row (Vision-only (1)) shows the accuracy of the baseline, and the last row (Our full model) shows the proposed method. The proposed method outperformed the baseline in both datasets.

Fig. 5 (b) shows the tendency of failure caused by Vision-only (1), which tends to recognize any objects as real foods despite the given set of $\mathcal{Y}_{\text{unseen}}$ in the prompt. On the other hand, as shown in Fig. 5 (c), the proposed method increases the chance of recognizing replicas as replicas.

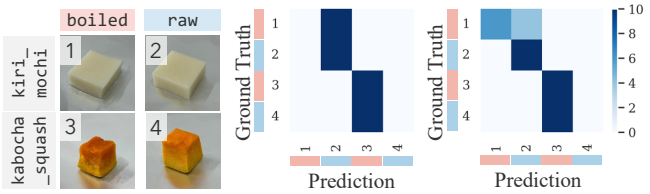
(a) the FoodReplica dataset



(b) vision only

(c) vision + tactile

Fig. 5. The FoodReplica dataset and the results. Class labels of replicas are given in the `resin_replica_${name}` format. The vision-only method predicted replicas as real in most cases, while the proposed (vision + tactile) method achieved a balanced performance.



(a) the Cube dataset (b) vision only (c) vision + tactile

Fig. 6. The Cube dataset and the results. Class labels are given in the `${state}_${name}` format (e.g., `boiled_kiri_mochi`). Tactile data helped to distinguish raw and boiled kiri_mochi, while kabocha_squash remained difficult to distinguish even with the proposed method.

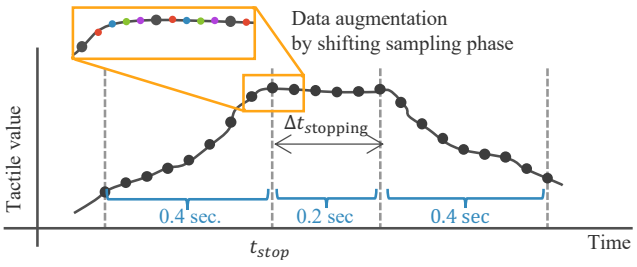


Fig. 7. Visualization of the data collection process. We crop the sequence based on the timing of t_{stop}, with simple data augmentation by shifting the offset for subsampling. This cropping operation results in $\boldsymbol{X}$ with $T = 25$.

This observation indicates that the proposed method can leverage the tactile data to complement the vision data. While overall performance is improved by using both vision and tactile sensors, some regression is observed in cases such as completely misclassifying a real canele as a replica canele and confusing a replica konjac with a replica chocolate. The misclassification of the canele could be due to the real canele being as hard as the replica, resulting in similar tactile properties. Similarly, the replica konjac and the replica chocolate may have been confused because they are tactilely similar when pressed from the side, and their appearances were also similar.

Fig. 6 (b) and (c) show the confusion matrices of the baseline and the proposed method on the Cube dataset. Both methods struggled to classify kabocha squash correctly, while

```
# Instruction
Identify the label of the object in the Image.
Follow the steps below to make your prediction:
1. Visual Analysis: Assign a score (0-10) based on
how well the image matches each label's expected
appearance.
2. Label Selection: The label with the highest
score is chosen as the correct one.

# Label options
${test_time_classes}

# Output Format
```
label - visual
{label} - {visual score} # Repeat for each
option
Answer: {selected label}
```
```

Fig. 8. Prompt for vision-only baselines. The prompt is the same as Fig. 3 except that expressions related to the tactile data are removed.

TABLE I

THE CLASSIFICATION ACCURACY ON FOODREPLICA AND CUBE.

Name	Method Tactile	top-k	VLM	Accuracy (%) FoodRep.	Cube
Vision-only (1)	–	–	GPT-4V	53.6	50.0
Vision-only (2)	–	–	LLaVA	39.0	25.0
LLaVA variant	VQ-VAE	3	LLaVA	39.1	25.0
β-VAE variant	β-VAE	3	GPT-4V	52.5	50.0
top-1 variant	VQ-VAE	1	GPT-4V	58.4	52.5
Our full model	VQ-VAE	3	GPT-4V	**58.9**	**65.0**

the proposed method improved the accuracy for boiled kiri mochi. Although the visual appearance of the boiled kiri mochi is quite similar to the raw one, the tactile properties are clearly different. As a result, many of the *description* of the boiled kiri mochi were like "otedama" and "kiwi_piece," which are soft objects, while those of the raw kiri mochi were like "wood_cube" and "stainless_cube," which are hard objects. It seems that these tactile textual descriptions helped the VLM to recognize the boiled kiri mochi correctly. Meanwhile, even the proposed method completely misclassified the raw kabocha squash as the boiled one. This could be because the tactile signals of the raw and boiled kabocha squash were similar and our method could not distinguish them. In this study, we defined the movement of the gripper as a soft touch to prevent the objects from breaking. However, the pressure was not enough to distinguish the raw and boiled kabocha squash. In future work, we need to consider the motion of the gripper to obtain more distinguishable tactile signals.

B. Comparison with the variants of the proposed method

We compared the proposed method with its variants, as shown in Tab. I. The first variant is the baseline method with GPT-4V replaced by LLaVA [3] (Vision-only (2)) and the second variant is the proposed method, which also uses LLaVA (LLaVA variant). The third variant is the proposed method incorporating the reparameterization trick of the β-VAE [36] instead of the vector quantization layer (β-VAE variant). The fourth variant is the proposed method but using only the top-1 nearest neighbor to organize *description*.

Tab. I shows that GPT-4V significantly outperforms LLaVA in this task. In addition, LLaVA hardly responded to tactile information in the prompt, which resulted in no performance increase. In contrast, our method improved the performance well with GPT-4V under the same conditions. For tactile encoding, VQ-VAE (our full model) outperformed the β-VAE variant. Moreover, the β-VAE variant performed worse than the baseline. This indicates that the vector quantization layer constructs a better tactile embedding space that improves classification accuracy compared to the reparameterization trick of the β-VAE. Finally, the top-3 implementation (our full model) outperformed the top-1 variant. This indicates that three references could better identify the property of unseen objects. From the aspect of absolute accuracy, we confirmed that there is still a large room for improvement in this challenging task.

Fig. 9. Accuracy of the proposed method with a controlled number of reference classes on the FoodReplica dataset. For the vision-only case and the full case, we tested the model once. For the other cases, we tested the model five times with different random combinations of training datasets and references. The plot shows the mean as a label and the standard deviation as an error bar for each case.

C. Analysis of the number of reference classes

Fig. 9 analyzes the impact of the number of reference classes in the training dataset and tactile-to-text database on the recognition performance. We obtained the results with $|\mathcal{Y}_{\text{ref}}| = 0$ (vision-only), 6, 13, 20, and 27 (full) on the FoodReplica dataset. Here, we did not include the validation data in the tactile-to-text database to control the parameter systematically.

We repeated the experiment five times with different combinations of training datasets and references, except for the vision-only case ($|\mathcal{Y}_{\text{ref}}| = 0$) and the full case ($|\mathcal{Y}_{\text{ref}}| = 27$). We could not find a clear trend in the accuracy, but the proposed method with the full training dataset achieved the highest accuracy. It indicates that the proposed method can leverage the tactile data more effectively with the full training dataset.

VI. CONCLUSIONS

We proposed a method for visuo-tactile zero-shot object recognition leveraging the common sense of VLMs. We constructed a tactile embedding network to extract a tactile embedding from the tactile sequence and a tactile-to-text database to convert the tactile embedding to a textual description. These were constructed from only the tactile sequences and class labels, eliminating the need for manual semantic labels and visual images bound to the tactile data. We used GPT-4V as the VLM to perform zero-shot object recognition using vision and tactile textual data. We evaluated the proposed method on the FoodReplica and Cube datasets and compared it with the vision-only baseline and its variants. The proposed method outperformed the baseline in both datasets. Future work includes exploring the motion of the gripper to obtain more distinguishable tactile signals and considering the best practices for prompt design to incorporate the tactile data more effectively into various VLMs.

ACKNOWLEDGMENT

The authors thank Reina Ishikawa for her invaluable support with robot operations.

REFERENCES

[1] R. S. Dahiya, G. Metta, M. Valle *et al.*, "Tactile sensing—from humans to humanoids," *IEEE Trans. on Robotics*, vol. 26, no. 1, pp. 1–20, 2010.

[2] R. W. Fleming, "Visual perception of materials and their properties," *Vision Research*, vol. 94, pp. 62–75, 2014.

[3] H. Liu, C. Li, Q. Wu *et al.*, "Visual instruction tuning," in *NeurIPS*, 2023, pp. 34 892–34 916.

[4] Z. Yang, L. Li, K. Lin *et al.*, "The dawn of LMMs: Preliminary explorations with GPT-4V(ision)," *arXiv preprint arXiv:2309.17421*, 2023.

[5] S. S. Kalakonda, S. Maheshwari, and R. K. Sarvadevabhatla, "Action-GPT: Leveraging large-scale language models for improved and generalized action generation," 2023, pp. 31–36.

[6] Z. Zhao, W. S. Lee, and D. Hsu, "Large language models as commonsense knowledge for large-scale task planning," in *NeurIPS*, 2023, pp. 31 967–31 987.

[7] J. Zhang, J. Huang, S. Jin, and S. Lu, "Vision-language models for vision tasks: A survey," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 46, no. 8, pp. 5625–5644, 2024.

[8] A. Radford, J. W. Kim, C. Hallacy *et al.*, "Learning transferable visual models from natural language supervision," *arXiv preprint arXiv:2103.00020*, 2021.

[9] Y. Gao, L. A. Hendricks, K. J. Kuchenbecker *et al.*, "Deep learning for tactile understanding from visual and haptic data," in *ICRA*, 2016, pp. 536–543.

[10] H. Liu, Y. Yu, F. Sun *et al.*, "Visual-tactile fusion for object recognition," *IEEE Trans. on Automation Science and Engineering*, vol. 14, no. 2, pp. 996–1008, 2017.

[11] B. Li, J. Bai, S. Qiu *et al.*, "VITO-Transformer: A visual-tactile fusion network for object recognition," *IEEE Trans. on Instrumentation and Measurement*, vol. 72, pp. 1–10, 2023.

[12] H. Liu, F. Sun, B. Fang *et al.*, "Cross-modal zero-shot-learning for tactile object recognition," *IEEE Trans. on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 7, pp. 2466–2474, 2020.

[13] Y. Fang, X. Zhang, W. Xu *et al.*, "Bidirectional visual-tactile cross-modal generation using latent feature space flow model," *Neural Networks*, vol. 172, p. 106088, 2024.

[14] F. Yang, C. Feng, Z. Chen *et al.*, "Binding touch to everything: Learning unified multimodal tactile representations," *arXiv:2401.18084*, 2024.

[15] Y. Li, J.-Y. Zhu, R. Tedrake *et al.*, "Connecting touch and vision via cross-modal prediction," in *CVPR*, 2019, pp. 10 601–10 610.

[16] K. Takahashi and J. Tan, "Deep visuo-tactile learning: Estimation of tactile properties from images," in *ICRA*, 2019, pp. 8951–8957.

[17] S. Cai, K. Zhu, Y. Ban *et al.*, "Visual-tactile cross-modal data generation using residue-fusion gan with feature-matching and perceptual losses," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7525–7532, 2021.

[18] Z. Abderrahmane, G. Ganesh, A. Crosnier, and A. Cherubini, "Visuo-tactile recognition of daily-life objects never seen or touched before," in *ICARCV*, 2018, pp. 1765–1770.

[19] L. Fu, G. Datta, H. Huang *et al.*, "A touch, vision, and language dataset for multimodal alignment," 2024.

[20] V. Chu, I. McMahon, L. Riano *et al.*, "Robotic learning of haptic adjectives through physical interaction," *Robotics and Autonomous Systems*, vol. 63, pp. 279–292, 2015.

[21] W. Hassan, J. B. Joolee, and S. Jeon, "Establishing haptic texture attribute space and predicting haptic attributes from image features using 1D-CNN," *Scientific Reports*, vol. 13, no. 1, p. 11684, 2023.

[22] A. Schmitz, Y. Bansho, K. Noda *et al.*, "Tactile object recognition using deep learning and dropout," in *ICHR*, 2014, pp. 1044–1050.

[23] Z. Abderrahmane, G. Ganesh, A. Crosnier *et al.*, "Haptic zero-shot learning: Recognition of objects never touched before," *Robotics and Autonomous Systems*, vol. 105, pp. 11–25, 2018.

[24] W. Yuan, C. Zhu, A. Owens *et al.*, "Shape-independent hardness estimation using deep learning and a GelSight tactile sensor," in *ICRA*, 2017, pp. 951–958.

[25] D. S. Chathuranga, Z. Wang, Y. Noh *et al.*, "Robust real time material classification algorithm using soft three axis tactile sensor: Evaluation of the algorithm," in *IROS*, 2015, pp. 2093–2098.

[26] S. S. Baishya and B. Bäuml, "Robust material classification with a tactile skin using deep learning," in *IROS*, 2016, pp. 8–15.

[27] T. Taunyazov, H. F. Koh, Y. Wu *et al.*, "Towards effective tactile identification of textures using a hybrid touch approach," in *ICRA*, 2019, pp. 4269–4275.

[28] W. Yuan, Y. Mo, S. Wang *et al.*, "Active clothing material perception using tactile sensing and deep learning," in *ICRA*, 2018, pp. 4842–4849.

[29] Z. Wang, S. Hirai, and S. Kawamura, "Challenges and opportunities in robotic food handling: A review," *Frontiers in Robotics and AI*, p. 433, 2022.

[30] M. C. Gemici and A. Saxena, "Learning haptic representation for manipulating deformable food objects," in *IROS*, 2014, pp. 638–645.

[31] A. Sawhney, S. Lee, K. Zhang *et al.*, "Playing with food: Learning food item representations through interactive exploration," in *ISER*, 2021, pp. 309–322.

[32] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *NeurIPS*, 2014, pp. 3104–3112.

[33] A. van den Oord, O. Vinyals, and K. Kavukcuoglu, "Neural discrete representation learning," in *NeurIPS*, 2017, pp. 6309–6318.

[34] Robotis, "OpenMANIPULATOR-X e-manual," https://emanual.robotis.com/docs/en/platform/openmanipulator_x/overview/, (Accessed on 15/03/2024).

[35] XELA Robotics, "Model XR1944," https://xelarobotics.com/xr1944, (Accessed on 29/06/2021).

[36] I. Higgins, L. Matthey, A. Pal *et al.*, "beta-vae: Learning basic visual concepts with a constrained variational framework," in *ICLR*, 2016.