Smartphone Contact-Object Estimation by Acoustic Sensing Focusing on Abstraction Level*

Haruya Nishi[1], Shigemi Ishida[1], Tomoki Murakami[2], and Shinya Otsuki[2]

[1] Future University Hakodate, Hokkaido, 041-8655 JAPAN
{g2123046,ish}@fun.ac.jp
[2] Access Network Service Systems Laboratories, Nippon Telegraph and Telephone Corporation, Japan

Abstract. Searching for the smartphone lost in a house is a time-consuming task because we usually rely on a ringing sound as a target signal. To support the smartphone search lost in a house, we are developing a smartphone search assistant system that estimates the smartphone's surrounding conditions based on acoustic sensing with a smart speaker. In this paper, we focus on smartphone contact-object estimation. Several studies have reported smartphone contact-object estimation using supervised machine learning (ML). However, the ML-based contact-object estimation fails when a smartphone is on an unknown object. There are too many objects in a house, which makes it impractical to train the object estimation model with all the objects in a house. Therefore, we propose a smartphone contact-object estimator that considers the abstraction level of estimation results to support unknown objects. Our estimator is based on two key ideas: (1) We prepare for estimator neural networks for multiple abstraction levels and switch the neural network model to a higher abstraction level when the estimation is unconfident. (2) We train the neural networks using information derived from the neural network corresponding to other abstraction levels. Experimental evaluation revealed that our proposed contact-object estimator successfully estimated a contact object with an accuracy of 0.991.

Keywords: Acoustic sensing · untrained object estimation · hierarchical neural network

1 Introduction

Many smartphone users often lose their smartphones in their houses [1]. We usually search for a lost smartphone relying on a ringing sound as a target signal by making a call to the lost smartphone from another device, which is inefficient due to the dependence on the human senses. When the lost smartphone is covered by something or is under something, a smartphone search might be more difficult.

* This work was supported in part by the Japan Society for the Promotion of Science (JSPS) KAKENHI Grant Numbers JP21K11847 and JP20KK0258 as well as the Cooperative Research Project Program of RIEC, Tohoku University.

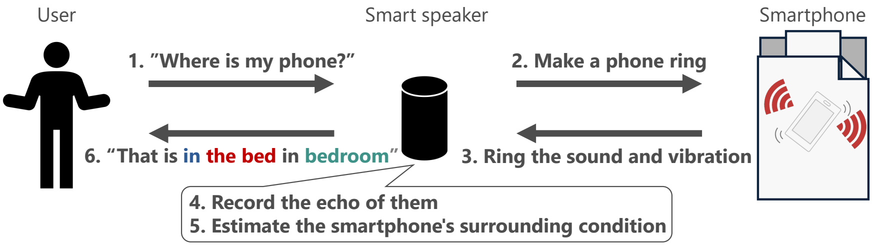

User
Smart speaker
Smartphone
1. "Where is my phone?"
2. Make a phone ring
3. Ring the sound and vibration
4. Record the echo of them
5. Estimate the smartphone's surrounding condition
6. "That is in the bed in bedroom"

Fig. 1: Overview of smartphone search assistance system

We are developing a smartphone search assistance system that employs acoustic sensing to estimate the smartphone's surrounding conditions using a smart speaker [2]. Figure 1 shows an overview of the smartphone search assistance system using a smart speaker. We define the smartphone's surrounding conditions as the room where the smartphone exists, the contact-object, and the cover state. Users can feel easier to find the lost smartphone with the smartphone's surrounding condition information.

In our previous work, we presented a smartphone cover-state classification method, which estimates one of the smartphone's surrounding conditions [2]. In this paper, we present a smartphone contact-object estimator, i.e., the second one of the smartphone's surrounding conditions.

Several studies have reported smartphone contact-object estimators using a supervised machine learning (ML) model. However, these estimators need to learn almost all objects in a house for smartphone search because they cannot handle untrained objects. Untrained objects are always mistakenly estimated as one of the trained objects. Estimation mistakes confuse a user to find a lost smartphone.

In contrast, we present a smartphone contact-object estimator that considers the abstraction level of estimation results to support unknown objects. Even if the object estimation fails, more abstract descriptions such as *the smartphone is on clothing* can be a hint to search for the lost smartphone. Our estimator switches the estimation model to one of a higher abstraction level to provide a hint for the search on estimation failures. The estimation failures are detected based on the confidence of an estimation result derived from the ML model.

Specifically, our contact-object estimator employs two approaches to improve the estimation and generalization performance: (1) We prepare for multiple neural network estimation models corresponding to abstraction levels and switch the model based on the estimation confidence. (2) We build a hierarchical neural network to share information among estimation models of different abstraction levels during model training. In this paper, we define three abstraction levels: object, material, and soft-hard levels.

To verify the effectiveness of our estimator, we evaluated the estimation and generalization performance of the above two approaches using data collected in a practical environment. The results show that our estimator effectively estimated contact objects with an estimation accuracy of 0.991 for 21 objects.

The rest of this paper is organized as follows. Section 2 describes related work of contact-object estimation and neural networks with a hierarchical structure. Section 3 describes our smartphone contact-object estimator that considers the abstraction level, followed by evaluation experiments in Sect. 4. Finally, Sect. 5 concludes this paper.

2 Related Work

2.1 Smartphone Contact-Object Estimation

To the best of our knowledge, this is the first attempt to estimate smartphone contact objects using a smart speaker. There have been smartphone contact-object estimators using smartphone built-in sensors such as a microphone [3–5], an accelerometer [6], a camera [7], and a combination of multiple sensors [8].

The microphone-based approach estimates a contact object using variation sound based on the acoustic characteristics of the contact object. Hwang et al. [3] estimated 12 contact objects such as a clothing pocket, desk, and chair with an accuracy of 0.910 based on the vibration sound difference of the contact objects. Ali et al. [5] also estimated 24 contact objects that usually exist both in a work office and home with an accuracy of 0.865 with the considerations of the effect of background noise. Hasegawa et al. [4] estimated 18 contact objects such as a clothing pocket, a wooden desk, and a smartphone stand with an accuracy of 0.821 based on the high-frequency components of the echo of a phone's beep sound.

The accelerometer-based approach estimates a contact object using the smartphone vibration characteristics affected by contact objects. Cho et al. [6] estimated six contact objects such as a sofa, bag, and hand with an accuracy of 0.850 based on the movement of a smartphone, which moves largely on the smoother surface of contact objects.

The multiple sensor-based approach estimates a contact object using smartphone built-in sensors such as a microphone, accelerometer, and magnetic sensor. Darbar et al. [8] estimated 13 contact objects with an accuracy of 0.917 using a rule-based hierarchical inference model based on microphone, magnetic sensor, and proximity sensor.

Although these studies have successfully estimated contact objects, no considerations on untrained objects have been taken. In a practical environment, there are many candidates for a contact object. Training with all the candidates is impractical.

2.2 Hierarchical Neural Networks

Hierarchical neural networks have been reported mainly for image classification tasks to improve performance [9–15].

Wang et al. [9] proposed scene classification using two abstraction levels, i.e., instances and parts. Instances represent all objects except the background, while parts represent components within instances. For example, an image containing multiple people is subdivided into instances of each person, which has parts such as a head, arms, and chest, for scene classification. Image classification is performed using based on both instances and parts.

Novack et al. [10] aimed to improve classification accuracy for unknown images by leveraging existing label hierarchy information and an implicit semantic hierarchy based on zero-shot image classification utilizing GPT-3. The implicit semantic hierarchy assumes a hierarchical structure exists between classes, even if the hierarchical structure is not explicitly defined in the dataset.

These studies demonstrated that hierarchical neural networks trained with labels of hierarchical structure improved classification performance. In this paper, we represent objects at different abstraction levels and utilize a hierarchically structured neural network. We train the hierarchical neural networks considering the hierarchical object representation.

3 Smartphone Contact-Object Estimator Considering Abstraction Level

3.1 Approach

The primary idea of our method is to take *abstraction levels* of objects into contact-object estimation. Estimation mistakes confuse a user to find a lost smartphone. Our method outputs information with a higher abstraction level when estimation confidence is low. As shown in Fig. 2, we can describe an object at different abstraction levels. For example, *a button-down* is *clothing* and *a soft object*. Our estimator outputs clothing when the estimation result of *a button-down* is derived with low confidence.

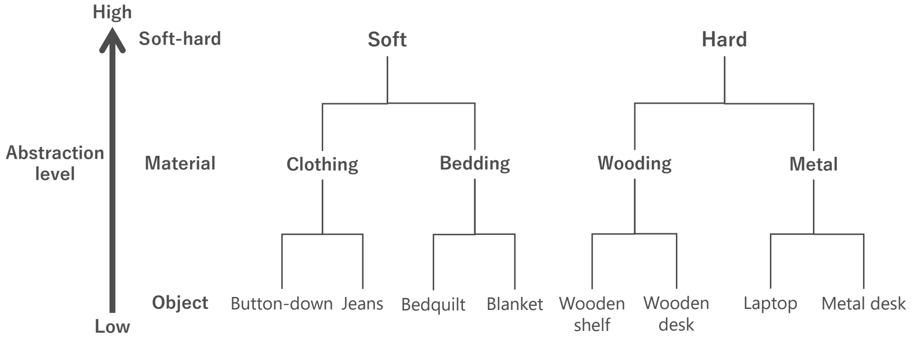


Fig. 2: Example of hierarchical representations of objects

In this paper, we define three abstraction levels: object, material, and soft-hard levels. We employ the following two approaches.

1. Estimation output with high confidence:
   We prepare for estimation models corresponding to abstraction levels. Our contact-object estimator calculates estimation confidence and marks estimation results invalid when estimation confidence is lower than a threshold. The final estimation output is the valid estimation result from the lowest abstraction level.
   In this paper, the lowest abstraction level is the object level, followed by material and soft-hard levels. When the estimation confidence at the object level is above the confidence threshold, for example, the estimation result at the object level is used as the final output. When the estimation confidence at the object and material levels are lower and higher than the threshold, respectively, the estimation result at the material level is the final output. The confidence threshold for each abstraction level is set on the training of the estimation model.
2. Information sharing between different abstraction levels on estimation model training:
   We use neural networks as an estimation model for each abstraction level. In the estimation model training, we transfer a part of the neural network model to another estimation model corresponding to different abstraction levels. In this way, we can include the feature extraction layers of the estimation models at different abstraction levels as part of the neural network, improving estimation accuracy. This approach is based on an intuition that we can estimate *a button-down* easier when we know the object is *clothing.*

3.2 Design overview

Figure 3 shows an overview of our smartphone contact-object estimator considering abstraction level. Our contact-object estimator consists of three blocks: a data collector, feature extractor, and hierarchical estimator. The data collector collects sound signals caused by smartphone vibration using a smart speaker built-in microphone. The feature extractor converts the sound signals into a mel-spectrogram, which is used as a feature vector for contact-object estimation as a classification task in the contact-object estimator. The hierarchical estimator consists of three estimation neural network models corresponding to three abstraction levels. The estimation result is taken from the results of the three models based on estimation confidence described as C_i in Fig. 3.

The neural network models in the hierarchical estimator share some layers. During the model training, we transfer these layers to share information trained for contact-object estimation at each abstraction level.

The following subsections describe the details of each block.

3.3 Data Collector

The data collector collects sound signals caused by smartphone vibration using a microphone embedded in a smart speaker.

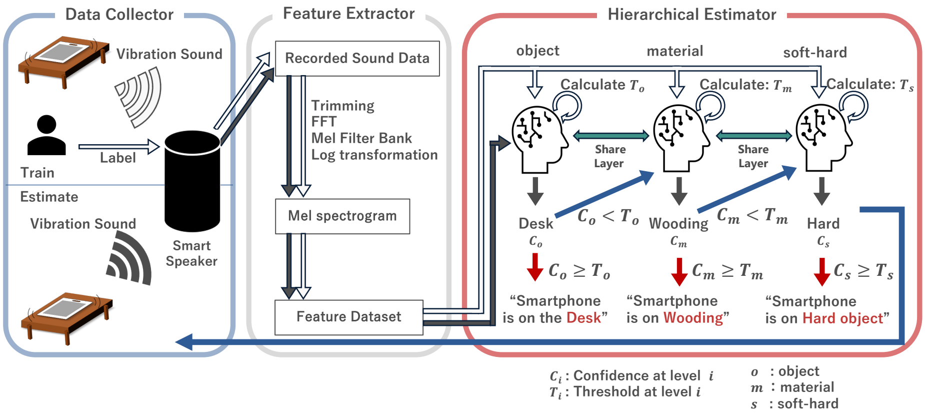


Fig. 3: Overview of smartphone contact-object estimator considering abstraction level

Sound signals used for the estimation model training are to be collected in daily life before a smartphone is lost. We assume that a smart speaker is connected to the smartphone, where we install a data collection application. The smart speaker collects vibration sound on everyday notifications. The smart speaker records vibration sound for 3 seconds immediately after the start of vibration, which is the same procedure presented in [5]. The data collection application then asks a user about the smartphone contact object to collect training label.

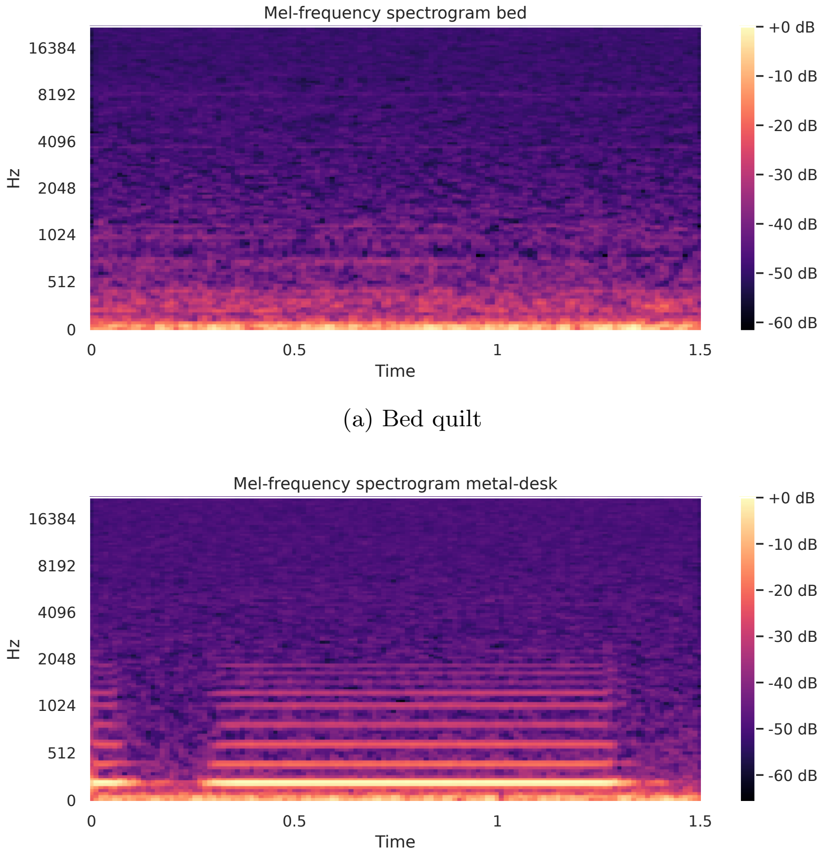
Sound signals used for the contact-object estimation are collected when a user asks the smart speaker to find a smartphone. The smart speaker sends a command to the smartphone to vibrate and collects the vibration sound.

3.4 Feature Extractor

The feature extractor calculates a mel-spectrogram from the recording data passed from the data collector.

First, we extract vibration sound data from the recording data. Because the starts of the recording and vibration are not precisely synchronized, the feature extractor trims off the first part of the recording data. The feature extractor extracts sound data between 100 and 1600 milliseconds from recording data, obtaining 1500-millisecond vibration sound data. The vibration length depends on the smartphone and might be less than 1500 milliseconds. If the vibration length is less than 1500 milliseconds, the feature extractor trims off non-vibration sound sections and repeats the vibration sound, obtaining the 1500-millisecond sound data.

Next, the feature extractor applies fast Fourier transform (FFT), mel filter bank, and logarithmic transformation to the vibration sound data to derive a mel-spectrogram in a logarithmic scale. Typical examples of mel-spectrograms



(a) Bed quilt

(b) Metal desk

Fig. 4: Example of mel spectrograms of different contact objects

are shown in Fig. 4. Figure 4 shows mel-spectrograms when the contact objects are bedquilt and metal desk. The mel-spectrogram represents the sound power in each frequency band at each time. We can see that the mel-spectrograms depend on the contact object.

In this paper, the feature extractor calculates a mel-spectrogram from the 1500-millisecond data sampled at 44.1 kHz. We use an FFT window size of 2048 and shift the window with an overlap size of 512. The number of channels in the mel-filter bank is 128. We obtain the sound power information for each FFT window, resulting in a mel-spectrogram dimension of 128×130.

3.5 Hierarchical Estimator

The hierarchical estimator estimates a smartphone contact object based on a mel-spectrogram obtained in the feature extractor. The hierarchical estimator

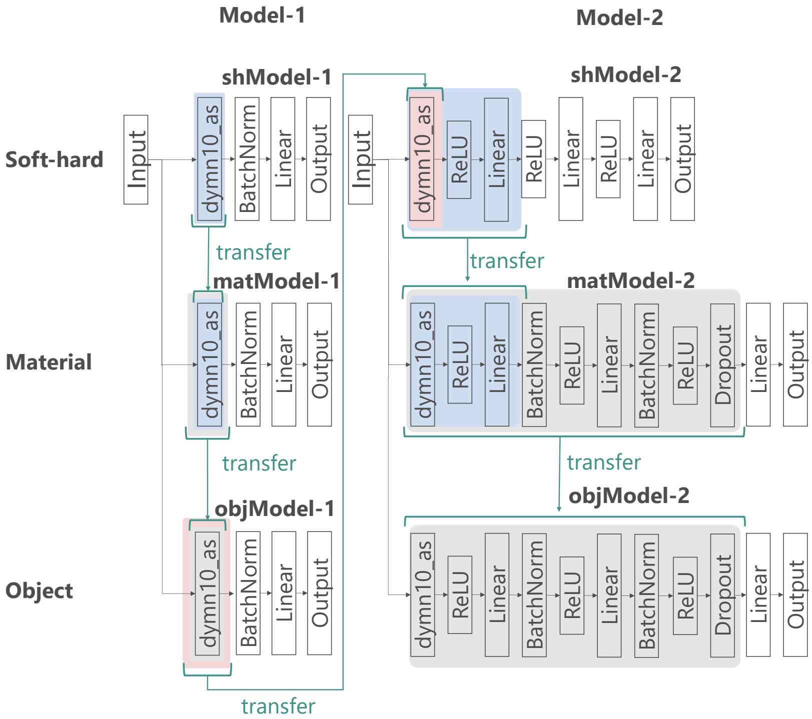


Fig. 5: Overview of the structure and training of neural network in hierarchical estimator

consists of estimation models corresponding to each abstraction level. Each estimation model outputs an estimation result and the estimation confidence of the contact object at a specific abstraction level.

Figure 5 shows the structure and the training overview of neural networks used in the hierarchical estimator. The neural network consists of Model-1 and Model-2, each of which includes estimation models corresponding to each abstraction level. Model-2 is used for contact-object estimation at each abstraction level, while Model-1 is an intermediate model used to train Model-2. Model-1 is discarded after the training.

The neural networks are trained in two rounds. In the first round, we train neural network models in Model-1 from the higher abstraction level. The model information at the higher abstraction level is transferred to the model at the lower abstraction level in the training. In the second round, we train neural network models in Model-2 from the higher abstraction level. We train the models with fine-tuning based on Model-1 or the higher estimation model in Model-2.

The actual training procedure is below. In this procedure, estimation models at soft-hard, material, and object levels in Model-i are represented as shModel-i, matModel-i, and objectModel-i, respectively. We use the dymn10-as deep neural network model from Schmid et al. [16] as a pre-trained model. dymn10-as is the fine-tuned ImageNet architecture trained with acoustic event dataset AudioSet.

1. shModel-1 training:
   We train the shModel-1 that consists of dymn10-as, BatchNorm, and Linear, i.e., Fully-Connected, layers.
2. matModel-1 training:
   We transfer the dymn10-as in the shModel-1 in this step. We extract the dymn10-as in the shModel-1 and append new BatchNorm and Linear layers, building a matModel-1 model. The matModel-1 is then trained.
3. objModel-1 training:
   We transfer the dymn10-as in the matModel-1 in this step. We extract the dymn10-as in the matModel-1 and append new BatchNorm and Linear layers, building an objModel-1 model. The objModel-1 is then trained.
4. shModel-2 training:
   We transfer the dymn10-as in the objModel-1 in this step. We extract the dymn10-as in the objModel-1 and append multiple new ReLU-Linear layers, building a shModel-2 model. The shModel-2 is then trained.
5. matModel-2 training:
   We transfer the feature extraction layers in the shModel-2, i.e., dymn10-as with ReLU and Linear layers, in this step. We extract the feature extraction layers in the shModel-2 and append new BatchNorm, ReLU, and Linear layers, building a matModel-2 model. The matModel-2 is then trained.
6. objModel-2 training:
   We transfer the matModel-2 except output layer, i.e., dymn10-as followed by BatchNorm, ReLU, and Linear layers, in this step. We removed the matModel-2's output layers and append new Linear layer as an output layer, building a objModel-2 model. The objModel-2 is then trained.

After the model training is completed, we determine confidence thresholds for each estimation model in Model-2. As shown in Fig. 3, the confidence thresholds are set at each abstraction level. We first input all the training data into each estimation model in Model-2, deriving estimation results and estimation confidence. We then filter incorrect results out. The confidence threshold is calculated for each estimation model as the mean of the confidence values corresponding to the remaining, i.e., correct, estimation results.

In this paper, we define the estimation confidence as the maximum output value of the Linear layer, i.e., the final output layer of each model in Model-2. The output of the linear layer is a set of real numbers for each class. For example, because the material-level estimation is a seven-class classification task, the linear-layer output is expressed as $[-2.02, -1.53, 0.45, 5.71, 0.05, -0.42, 1.05]$. In this example, the estimation confidence is the maximum output of 5.71.

The hierarchical estimator chooses the estimation results of Model-2 as an overall contact-object estimation result based on the estimation confidence at each abstraction level. Let T_s, T_m, T_o be the confidence threshold at soft-hard, material, and object levels, respectively, and C_s, C_m, C_o be the estimation confidence values at soft-hard, material, and object levels, respectively. The output of the hierarchical estimator is determined as:

1. When $C_o \geq T_o$:
   The hierarchical estimator outputs the objModel-2 estimation result.
2. When $C_m \geq T_m$:
   The hierarchical estimator outputs the matModel-2 estimation result.
3. When $C_s \geq T_s$:
   The hierarchical estimator outputs the shModel-2 estimation result.
4. When $C_s < T_s$:
   The hierarchical estimator retires the overall estimation process from the beginning. The data collector collects vibration sound data again, followed by feature extraction and object estimation. If $C_s < T_s$ in the re-estimation again, the hierarchical estimator outputs *unknown*.

When the output of the hierarchical estimator is *unknown*, the smartphone search assistance system, shown in Fig. 1, provides information only about the cover state and the smartphone-located room.

4 Evaluation

We evaluated our smartphone contact-object estimator using the data collected in a practical environment. We first evaluated the estimation accuracy of the estimation models of each abstraction level as a micro-level evaluation. We then evaluated the overall contact-object estimation accuracy, i.e., smartphone contact-object estimation based on estimation confidence, as a macro-level evaluation. Finally, the estimation accuracy against untrained objects was evaluated.

4.1 Experiment Setup

Figure 6 shows the data collection experiment setup. A target contact object was put on a metal desk with a height of approximately 70 centimeters. We put an ASUS Zenfone 8 smartphone face up on the target object and installed an audio-technica AT2050 microphone approximately 1 meter away from the object at a height of approximately 70 centimeters. We set the directionality of the microphone toward the smartphone. The microphone was connected to a ZOOM H6 audio recorder. Note that the oversized target objects that could not be placed on the metal desk were directly placed on a wooden floor.

Figure 7 shows the list of target contact objects in this experiment. We used 21 contact objects, which can be described in 7 materials. Based on the material, we assigned a label of soft or hard to each object.

We collected vibration sound data in the following procedure. The sound data was collected while changing the position and orientation of the smartphone for each trial to emulate different contact conditions.

1. Place the smartphone on the contact object.
2. Start recording.
3. Activate vibration for one second.

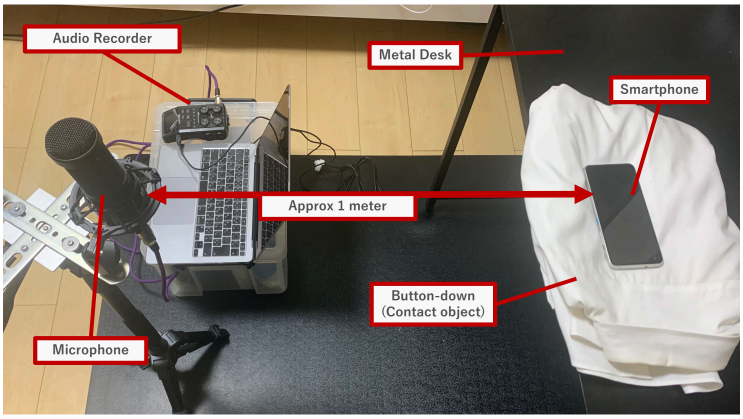
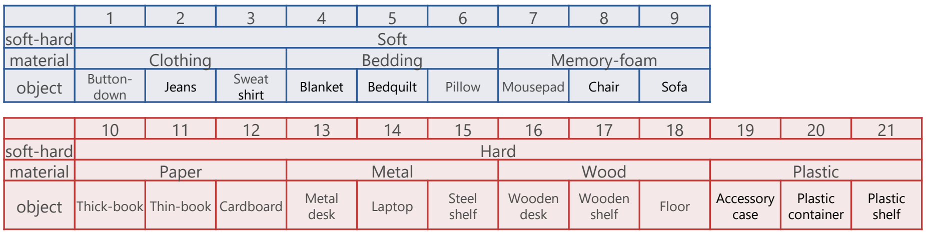


Fig. 6: Experiment setup



	1	2	3	4	5	6	7	8	9
soft-hard	Soft								
material	Clothing			Bedding			Memory-foam		
object	Button-down	Jeans	Sweat shirt	Blanket	Bedquilt	Pillow	Mousepad	Chair	Sofa

	10	11	12	13	14	15	16	17	18	19	20	21
soft-hard	Hard											
material	Paper			Metal			Wood			Plastic		
object	Thick-book	Thin-book	Cardboard	Metal desk	Laptop	Steel shelf	Wooden desk	Wooden shelf	Floor	Accessory case	Plastic container	Plastic shelf

Fig. 7: Target contact objects in this experiment

4. Take the smartphone off from the contact object and place the smartphone again to change the position and orientation of the smartphone.
5. Repeat steps 3 and 4 for 50 times.
6. Stop recording.

In total, we collected the recording data of 50 trials for each object. At the beginning of each trial, we made a sound of 2000 Hz pure tone for 100 milliseconds as a trial onset marker. Referring to the trial onset markers, we split the recording data for each trial.

To demonstrate the effectiveness of the proposed contact-object estimator, we compared the estimation accuracy between the following two methods. We calculated the F-score for each label and calculated the harmonic mean of the F-scores for all labels, deriving the estimation accuracy.

1. Proposed method
   The method presented in Sect. 3. For the proposed method, we evaluated the F-score with a hold-out method, which randomly splits the dataset into training and test datasets only once, because of the long training time.
2. SVM method

Table 1: Estimation accuracy at each abstraction level

Abstraction level	SVM	Model-1	Model-2
Object	0.843	0.824	0.857
Material	0.850	0.714	0.876
Soft-hard	0.881	0.890	0.938

The smartphone contact-object estimation method presented in our previous work [2]. The SVM method uses a support vector machine (SVM) classifier with mel-frequency cepstral coefficients (MFCCs) as features instead of a mel-spectrogram. We prepared for estimation models for each abstraction level and separately trained the estimation models. We evaluated the SVM method with the harmonic mean of estimation accuracies derived with 10-fold cross-validation.

4.2 Estimation Accuracy of the Estimation Models at Each Abstraction Level

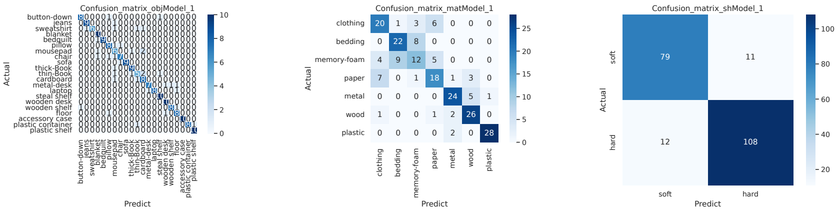
To evaluate the performance of the estimation models at each abstraction level, we compared the estimation accuracy of models in Model-1 and Model-2. The collected data were randomly sorted and split in the ratio of training : validation : test = 6 : 2 : 2. We then trained and evaluated the estimation models at each abstraction level.

Figures 8 and 9 show the confusion matrices of contact-object estimation results for each abstraction level using models in Model-1 and Model-2, respectively. From Fig. 8, we can see that Model-1 successfully estimated the contact objects at the object and material levels. On the other hand, there were incorrect estimations at the material level, especially for soft objects such as memory foam.

Referring to Fig. 9, we can confirm that Model-2 demonstrated a high degree of success in estimating objects at all abstraction levels. Particularly at the material level, Model-2 appears to have outperformed Model-1.

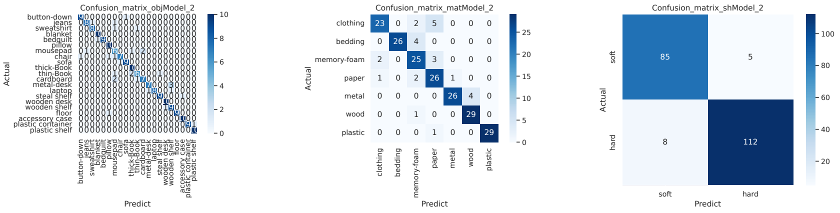
Table 1 shows the estimation accuracy of the estimation models at each abstraction level. The table also shows the estimation accuracy of the SVM method. As shown in Table 1, the proposed method, i.e., Model-2, showed the highest accuracy among the SVM, Model-1, and Model-2 at all the abstraction levels. At the material and soft-hard levels, Model-2 greatly improved the estimation accuracy compared to Model-1. The 2-round training successfully captured the information of other abstraction levels, resulting in higher accuracy.

The estimation accuracy of the models in Model-1 was lower than that of the SVM method at the material and soft-hard levels, as shown in Table 1. This was mainly caused by the limited amount of training data against the complicated neural network. The related studies [6] and [8] have also shown the difficulties in the recognition of soft and hard. Our 2-round training approach was effective in training the soft-hard classifier with the limited training data.



(a) Object level (b) Material level (c) Soft-hard level

Fig. 8: Confusion matrices of contact-object estimation results by Model-1



(a) Object level (b) Material level (c) Soft-hard level

Fig. 9: Confusion matrices of contact-object estimation results by Model-2

The above results confirm that the estimation model training using the information derived from other abstraction levels improved the estimation accuracy.

4.3 Overall Estimation Accuracy

To evaluate the overall performance of contact-object estimation, we evaluated the estimation accuracy of contact-object estimation based on estimation confidence. For each trial of the collected data, we obtained a final estimation output from the estimation results at each abstraction level derived in the previous subsection. The output of the hierarchical estimator was determined based on the estimation confidence according to the estimation procedure presented in Sect. 3.5. We then calculate the harmonic mean of the F-score for estimation results at each abstraction level.

Table 2 shows the overall estimation accuracy of the SVM and Model-2, i.e., the proposed method. The table also shows the estimation completion rate, which is the rate of test trials whose estimation confidence exceeded the confidence threshold. Table 2 shows that the estimation accuracy of the proposed method was higher than that of the SVM method at all the abstraction levels. Comparing the estimation accuracy of the SVM and the proposed method, we can confirm that the proposed method improved the overall estimation accuracy.

In Table 2, the estimation completion rate of the SVM and Model-2 at the object level was 43.5% and 51.4%, respectively. This result indicates that we

Table 2: Overall estimation accuracy and estimation completion rate

Abstraction level	Accuracy		Estimation completion rate	
	SVM	Model-2	SVM	Model-2
Object	0.967	0.991	43.5%	51.4%
Material	0.877	0.909	19.2%	10.5%
Soft-hard	0.954	1.000	15.4%	10.0%

derived a high accuracy for more number of test trials by the proposed method compared to the SVM method. The total estimation completion rate can be calculated by summing the estimation completion rates at all the abstraction levels. The total estimation completion rate of the SVM method and Model-2 was 78.1% and 71.9%, respectively. Both methods could estimate the contact object for more than 70% of the trials. At material and soft-hard levels, the estimation completion rate of SVM was higher than that of Model-2. Model-2 discarded estimation results at these abstraction levels when the confidence level was low, resulting in high accuracy.

Comparing Tables 1 and 2, we can confirm that the estimation accuracy for both SVM and Model-2 was improved at all the abstraction levels. The estimation output decision based on the estimation confidence successfully improved the estimation accuracy at each abstraction level, not depending on the estimation model.

The above results confirm that the contact-object estimation based on estimation confidence improved estimation accuracy.

4.4 Estimation Accuracy against Untrained Objects

Our final goal is to realize a contact-object estimator that supports untrained objects. We evaluated the contact-object estimation accuracy when the smartphone is on an untrained object. We performed Leave-One-Object-Out (LOOO) cross-validation in this evaluation. For each of the 21 objects, we used one object as test data and used the remaining data of 20 objects as training data to derive the harmonic mean of F-scores. Note that the training data was again randomly split in the ratio of training : validation = 8 : 2 for the neural network training.

First, we evaluated the estimation accuracy without the confidence-based decision. Table 3 shows the estimation accuracy of SVM, Model-1, and Model-2 against untrained objects without estimation confidence-based decision. We excluded the estimation accuracy at the object level because the object-level estimator always outputs an incorrect estimation result as we used supervised learning algorithms. The estimation accuracy of the proposed method, i.e., Model-2, was 0.521 and 0.872 at material and soft-hard levels, respectively, which were higher than those of the SVM of 0.330 and 0.812. We can confirm that Model-2's generalization performance was higher than that of the SVM at all the abstraction levels. Comparing the estimation accuracy between Model-2 and Model-1, we can also confirm that the Model-2's generalization performance was also

Table 3: Estimation accuracy of contact-object estimation for untrained objects without confidence-based decision

Abstraction level	SVM	Model-1	Model-2
Material	0.330	0.483	0.521
Soft-hard	0.812	0.813	0.872

Table 4: Overall estimation accuracy against untrained objects

Abstraction level	Accuracy		Estimation completion rate	
	SVM	Model-2	SVM	Model-2
Object	–	–	21.0%	7.8%
Material	0.338	0.580	20.0%	46.2%
Soft-hard	0.740	0.836	25.0%	5.8%

higher than that of Model-1. Our 2-round training with the hierarchical neural network improved the generalization performance and improved the object estimation performance against untrained objects.

Next, we evaluated the overall estimation accuracy against untrained objects. Table 4 shows the overall estimation accuracy, i.e., estimation accuracy with the confidence-level decision, against untrained objects. The estimation accuracy of Model-2, i.e., the proposed method, was 0.580 and 0.836 at the material and soft-hard levels, respectively, which were higher than those of the SVM of 0.338 and 0.740. We can confirm that Model-2's generalization performance was higher than that of the SVM method at all abstraction levels.

At the object level, the estimation completion rate of the proposed method was 7.8%, which is lower than that of the SVM method of 21.0%. For untrained objects, estimation completion at the object level means incorrect estimation. The lower completion rate of the proposed method at the object level indicates high robustness against untrained objects.

At the material level, the estimation completion rate of the proposed method was higher than that of the SVM method. The proposed method performs better in capturing features of hierarchical object representation, which resulted in the higher estimation completion rate for untrained objects.

On the other hand, at the soft-hard level, the estimation completion rate of the proposed method was lower than that of the SVM method. In total, estimation was completed for $7.8 + 46.2 + 5.8 = 59.8\%$ trials in the proposed method, while $21.0 + 20.0 + 25.0 = 66.0\%$ of estimation trials were completed in the SVM method. Note that estimation completion does not indicate correct estimation. In the proposed method, we sacrificed unsure estimation results, which resulted in high accuracy and low estimation completion rate. There might be room to improve the soft-hard estimation of the proposed method while keeping estimation accuracy.

The above results confirm that our contact-object estimator, employing a hierarchical neural network and considering abstraction level, improved the estimation accuracy against untrained objects. We think the estimation accuracy was still insufficient for practical use, especially at the material level. We are working to more improve estimation accuracy against untrained objects.

5 Conclusion

In this paper, we proposed a smartphone contact-object estimator for an in-home smartphone search support system. Machine learning-based contact-object estimators have been proposed, though, they have difficulties in estimation against untrained objects. We therefore proposed a contact-object estimator considering abstraction level to support untrained objects. The proposed method relies on two approaches: (1) We prepare hierarchical neural networks for multiple abstraction levels and switch the neural network to a higher abstraction level when the estimation is unconfident. (2) We train the neural network using information derived from other abstraction levels. We conducted experimental evaluations and confirmed that our contact-object estimator estimated contact objects with an accuracy of 0.991 for 21 objects, demonstrating the effectiveness of the above two approaches. We are planning to work on contact-object estimation considering other surrounding conditions such as a cover state.

References

1. TrackR: Survey on what you are looking for: https://prtimes.jp/main/html/rd/p/000000006.000022312.html (2013).
2. Nishi, H., Ishida, S., Murakami, T. and Otsuki, S.: Smartphone Cover-State Classification via Acoustic Sensing for Smartphone Search in Indoor Environments, *2023 IEEE 12th Global Conference on Consumer Electronics (GCCE)*, IEEE, pp. 429–430 (2023).
3. Hwang, S. and Wohn, K.: VibroTactor: Low-Cost Placement-Aware Technique Using Vibration Echoes on Mobile Devices, pp. 73–74 (2013).
4. Hasegawa, T., Hirahashi, S. and Koshino, M.: Determining Smartphone's Placement Through Material Detection, Using Multiple Features Produced in Sound Echoes, *IEEE Access*, Vol. 5, pp. 5331–5339 (2017).
5. Ali, K. and Liu, A. X.: Fine-Grained Vibration Based Sensing Using a Smartphone, *IEEE Transactions on Mobile Computing*, Vol. 21, No. 11, pp. 3971–3985 (2021).
6. Cho, J., Hwang, I. and Oh, S.: Vibration-Based Surface Recognition for Smartphones, *2012 IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, pp. 459–464 (2012).
7. Yeo, H.-S., Lee, J., Bianchi, A., Harris-Birtill, D. and Quigley, A.: SpeCam: Sensing Surface Color and Material with the Front-Facing Camera of a Mobile Device, *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services*, Vienna Austria, ACM, pp. 1–9 (2017).
8. Darbar, R. and Samanta, D.: SurfaceSense: Smartphone Can Recognize Where It Is Kept, *Proceedings of the 7th Indian Conference on Human-Computer Interaction*, IndiaHCI '15, New York, NY, USA, Association for Computing Machinery, pp. 39–46 (2015).
9. Wang, X., Li, S., Kallidromitis, K., Kato, Y., Kozuka, K. and Darrell, T.: Hierarchical Open-Vocabulary Universal Image Segmentation, *Advances in Neural Information Processing Systems*, Vol. 36, pp. 21429–21453 (2024).
10. Novack, Z., McAuley, J., Lipton, Z. C. and Garg, S.: Chils: Zero-shot Image Classification with Hierarchical Label Sets, *International Conference on Machine Learning*, PMLR, pp. 26342–26362 (2023).
11. Gargiulo, F., Silvestri, S. and Ciampi, M.: Exploit Hierarchical Label Knowledge for Deep Learning, *2019 IEEE 32nd International Symposium on Computer-Based Medical Systems (CBMS)*, pp. 539–542 (2019).
12. Giunchiglia, E. and Lukasiewicz, T.: Coherent Hierarchical Multi-Label Classification Networks, *Advances in neural information processing systems*, Vol. 33, pp. 9662–9673 (2020).
13. Maltoudoglou, L., Paisios, A., Lenc, L., Martínek, J., Král, P. and Papadopoulos, H.: Well-Calibrated Confidence Measures for Multi-Label Text Classification with a Large Number of Labels, *Pattern Recognition*, Vol. 122, p. 108271 (2022).
14. Wang, R., Ridley, R., Qu, W. and Dai, X.: A Novel Reasoning Mechanism for Multi-Label Text Classification, *Information Processing & Management*, Vol. 58, No. 2, p. 102441 (2021).
15. Zou, X., Yang, J., Zhang, H., Li, F., Li, L., Wang, J., Wang, L., Gao, J. and Lee, Y. J.: Segment Everything Everywhere All at Once, *Advances in Neural Information Processing Systems*, Vol. 36 (2024).
16. Schmid, F., Koutini, K. and Widmer, G.: Dynamic Convolutional Neural Networks as Efficient Pre-trained Audio Models, *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Vol. 32, pp. 2227–2241 (2024).