applied sciences

MDPI

Article

A Few-Shot Learning-Based Material Recognition Scheme Using Smartphones

Yeonju Kim, Jeonghyeon Yoon and Seungku Kim *

Department of Electronics Engineering, Chungbuk National University, Cheongju 28644, Republic of Korea; heyyjjj@gmail.com (Y.K.); thera9234@chungbuk.ac.kr (J.Y.)
* Correspondence: kimsk@cbnu.ac.kr

Abstract: This study proposes FSMR, a material recognition scheme designed to expand context information about locations in context recognition services. FSMR identifies the material in contact with the smartphone and determines the object based on this information to obtain location data. When the smartphone sends vibrations to the object it touches, the vibration signals change according to the unique properties of the material, and the reflected signals are measured using an accelerometer. Based on the fact that the measured sensor values have distinct characteristics for each material, deep learning techniques are applied to classify the material and determine the object. The existing research on material and object recognition using smartphone vibrations and accelerometers often requires vast amounts of training data for deep learning-based models, making it challenging to apply to real-world applications. To address this issue, this study employs few-shot learning and data augmentation to significantly reduce the amount of training data required. The evaluation results show that FSMR achieved classification accuracies of up to 72.03% and 83.63% when trained with data collected over 1 s and 5 s, respectively.

Keywords: context-aware; data augmentation; deep learning; few-shot learning; material recognition; smartphone

check for updates

Academic Editor: Christos Bouras

Received: 27 November 2024
Revised: 26 December 2024
Accepted: 1 January 2025
Published: 5 January 2025

Citation: Kim, Y.; Yoon, J.; Kim, S. A Few-Shot Learning-Based Material Recognition Scheme Using Smartphones. *Appl. Sci.* **2025**, *15*, 430. https://doi.org/10.3390/app15010430

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Recent advancements in digital technologies have focused on providing personalized services to enhance the user experience. One of the core concepts of such personalized services is context-aware services [1,2]. Context-aware services refer to identifying a user's current situation or surrounding environment and providing services tailored to that context. For example, when a user enters an indoor space, a smartphone can collect location and temperature data through GPS and sensors, recognize the situation, and automatically turn on the indoor lights or activate the air conditioner. These services, which combine various sensors, have the advantage of being applicable in multiple domains such as smart homes, healthcare, and smart cities. A representative example of a context-aware service utilizing smartphones is Bixby Routines. Bixby Routines is a smartphone application that enhances daily convenience by collecting information from various built-in sensors, inferring the user's context and providing personalized services. To provide users with appropriate context-aware services, accurate location information is essential. Location information is one of the most critical contextual elements and is indispensable for improving the accuracy and quality of services. If location information is missing or inaccurate, incorrect services may be provided to the user. Accurate location information not only enhances service quality but also enables a more sophisticated context inference when fused with other sensor data.

To deliver context-aware services based on detailed location information, material recognition techniques that distinguish the surfaces in contact with a smartphone using various sensors have been studied. J. Yang et al. [3] proposed a method for determining whether a smartphone is in a pocket or not by utilizing the distance and light between the smartphone and nearby objects. By setting thresholds for light and distance values measured by sensors, J. Yang et al. differentiated whether the smartphone was in a pocket, a bag, or another location. H. Yeo et al. [4] proposed SpeCam, a method that uses a smartphone camera to recognize the material and color of surfaces in contact with the smartphone. SpeCam utilizes the smartphone display and front camera, performing classification through machine learning. Technologies like these, which leverage light sensors and cameras on smartphones to identify location information, have been extensively researched to provide better services to users. However, existing methods that rely on light and distance have limitations in distinguishing various objects. These methods are influenced by the resolution and quality of cameras, observe only the exterior of nearby objects, and cannot differentiate objects with similar materials. Additionally, privacy concerns may arise with camera-based approaches.

To address these issues, this study proposes a material recognition technique that identifies the material of objects in contact with a smartphone by utilizing vibrations. When a smartphone vibrates, the vibration interacts differently depending on the unique physical properties of the contacting material, such as diffraction, attenuation, and reflection, enabling the differentiation of objects with similar materials. This method is less challenging in terms of classification compared to existing camera-based methods and does not pose privacy concerns. Previous studies have also explored material recognition techniques using vibrations [5,6], but these approaches often rely on deep learning models that require a large amount of data for training. Collecting such a volume of data is impractical for real-world users. Therefore, this study introduces FSMR (few-shot learning-based material recognition), a material recognition method for smartphones that requires minimal data by leveraging vibrations and accelerometers. FSMR employs few-shot learning and data augmentation techniques. By reducing the amount of training data needed to classify new materials not previously trained, FSMR significantly enhances usability for users.

The contributions of this paper are as follows:

1. Proposing a material recognition method based on few-shot learning, significantly reducing the amount of data required for training;
2. Introducing a data augmentation technique for smartphone vibration data, increasing the amount of data available for few-shot learning training and reducing the amount of data required to classify new materials.

The structure of this paper is as follows. Section 2 introduces related works, including various studies on context-aware services and detection systems utilizing smartphone vibrations. Section 3 discusses background knowledge, explaining the limitations of traditional deep learning-based methods that require extensive data and the concept of few-shot learning. Section 4 presents FSMR, the proposed method, detailing the data collection process, data augmentation techniques, and model design. Section 5 evaluates and experiments on FSMR, while Section 6 concludes the paper.

2. Related Work

2.1. Context-Aware Services

Smartphones have become essential personal devices in daily life due to their widespread supply since the 2010s. They are capable of interconnecting people and the internet while featuring data processing capabilities, making them one of the most critical Internet of Things (IoT) devices. Also, by utilizing the diverse features and built-in sensors

of smartphones, it is possible to collect context-specific data about users' surrounding environments and activities [7]. Such smartphone data serve as valuable resources for building rule-based intelligent context-aware systems to understand user behavior patterns in various situations [8].

Early context-aware services primarily focused on the relationship between people and devices to infer contexts. However, recent studies include diverse factors such as users' activities, device states, and app usage histories. Schilit et al. [9] considered contextual factors like user location, people and objects around the user, and changes in objects. Dey [10] defined context as "any information that affects the daily situations of users and influences smartphone usage", which can include temporal, spatial, and social contexts. Sarker et al. [11] described context-aware services as formalizing "IF–THEN" logic structures. The "IF" part represents contextual information about the user's surroundings, while the "THEN" part corresponds to the actions or states taken in response to that context. For example, in the case of "User A sets the smartphone to silent mode while at work", "workplace" is spatial contextual information, and "silent mode" represents the smartphone's state in that context. Such context-aware service rules and personalized services derived from large datasets can be applied in a wide range of fields [12].

With smartphones connecting all surrounding contexts as data sources, various services have been developed to predict users' smartphone usage in specific situations [13–16]. For instance, smartphones can store data about users' contexts when making calls, receiving calls, rejecting calls, or leaving calls unanswered. Contextual data such as the user's location [17] or the social relationship between the caller and receiver [18] can be used to provide call-related context-aware services.

2.2. Material Recognition Using Vibration

Xu et al. [19] proposed TouchPass, a system for user identification using smartphone vibrations and accelerometers. TouchPass is an authentication framework designed to prevent smartphone data breaches, enabling user authentication regardless of finger movement when the user touches the smartphone. Xu et al. demonstrated through the mathematical modeling of vibrations that unique physical properties of fingers, such as density and conductivity, can be distinguished. To separate the transient state and steady state of vibrations, they designed vibration signals that oscillate multiple times in short cycles. These signals were preprocessed to distinguish between transient and steady states, and features were extracted using frequency spectrum and cepstrum analysis. The extracted features were then reconstructed using a Siamese Network to ensure behavior-independent user differentiation and trained with a Knowledge Distillation-Based Classifier Training approach. S. Hwang et al. [5] proposed VibroTactor, a system that recognizes objects in contact with a smartphone using smartphone vibrations and microphones. VibroTactor is a cost-effective technology for determining the location of a smartphone (e.g., pocket, bed, desk, etc.). It filters and analyzes acoustic signals generated by smartphone vibrations to determine the device's location. Since it uses built-in microphones on commercial smartphones, VibroTactor enables location recognition at a low cost. Classification is performed using an RBF (Radial Basis Function) Network, achieving an average accuracy of 91% across 12 different locations. J. Cho et al. [6] proposed VibePhone, a system that identifies objects in contact with a smartphone using smartphone vibrations and accelerometers. It records accelerometer readings when the smartphone vibrates and analyzes vibration patterns based on the material of the object's surface. The system uses a Support Vector Machine (SVM)-based deep learning model to classify materials, improving accuracy through preprocessing and feature extraction techniques. It achieved an average accuracy of 85% across six labels: sofa, plastic table, wooden table, hand, backpack, and pants pocket. For

each sample, 1.5 s of vibration data were collected, and training required 120 labeled data points per class.

Although these studies achieved a high classification accuracy, they share a significant limitation: the requirement for large amounts of training data. For instance, VibePhone required 120 training samples per label, with each sample consisting of 1.5 s of vibration data. Additionally, surfaces not previously trained in the model could not be classified until sufficient data was collected and the model was retrained. These limitations reduce the practical applicability of such systems [20]. Therefore, a new approach is required that reduces the amount of data needed for training while maintaining high performance.

3. Background

The existing study [6], which utilizes smartphone vibrations and accelerometers for material recognition, is based on a deep learning model to identify the surface in contact with the smartphone. For this, the study required 180 s of accelerometer data per label. Such a lengthy data collection process makes it impractical for users to manually register labels in real-life applications. To enhance usability and reduce the need for extensive training data, the core technologies of FSMR (few-shot learning-based material recognition) are data augmentation [21] and few-shot learning. This section introduces the concept of few-shot learning.

Few-Shot Learning Method

Humans can learn from just a few examples, whereas deep learning models typically require hundreds of data points for effective training [22]. The concept of enabling deep learning models to learn new concepts like humans is called meta-learning, and learning with minimal examples is referred to as few-shot learning [23–28]. Few-shot learning aims to train classifiers to recognize new classes with only a small number of examples. When training involves K data points for N classes, the process is described as N-way K-shot learning. Figure 1 illustrates the three-way three-shot learning method in few-shot learning. Few-shot learning operates in task units, consisting of a support set and a query set, to achieve meta-learning. The dataset used for training is called the meta-training set, while the dataset used for evaluation is referred to as the meta-test set. For validation, the dataset is known as the meta-validation set. One notable few-shot learning model, Matching Networks, facilitates rapid learning. Deep learning models often require large datasets due to the need for iterative weight updates using Stochastic Gradient Descent (SGD). Matching Networks, however, leverages ideas from non-parametric models, such as K-Nearest Neighbors (KNN), which dynamically adjust to the data instead of predefining parameters. This allows the design of models that achieve excellent generalization performance with limited data. Matching Networks utilizes an attention mechanism to predict unlabeled query sets using embeddings learned from labeled support sets. It also employs mini-batches, called episodes, to achieve superior generalization performance.

Prototypical Networks (ProtoNet) [28] are another popular model designed for few-shot and zero-shot learning. Their primary focus is on addressing the overfitting problem. This model assumes that a highly simplified classification is necessary for learning with limited data. ProtoNet defines a single prototype for each class as a cluster in the embedding space. The support set is mapped into the embedding space through nonlinear transformations via neural networks, and the meaning of these mappings is used as the class prototype. Classification is then performed by identifying the prototype closest to a given query point.

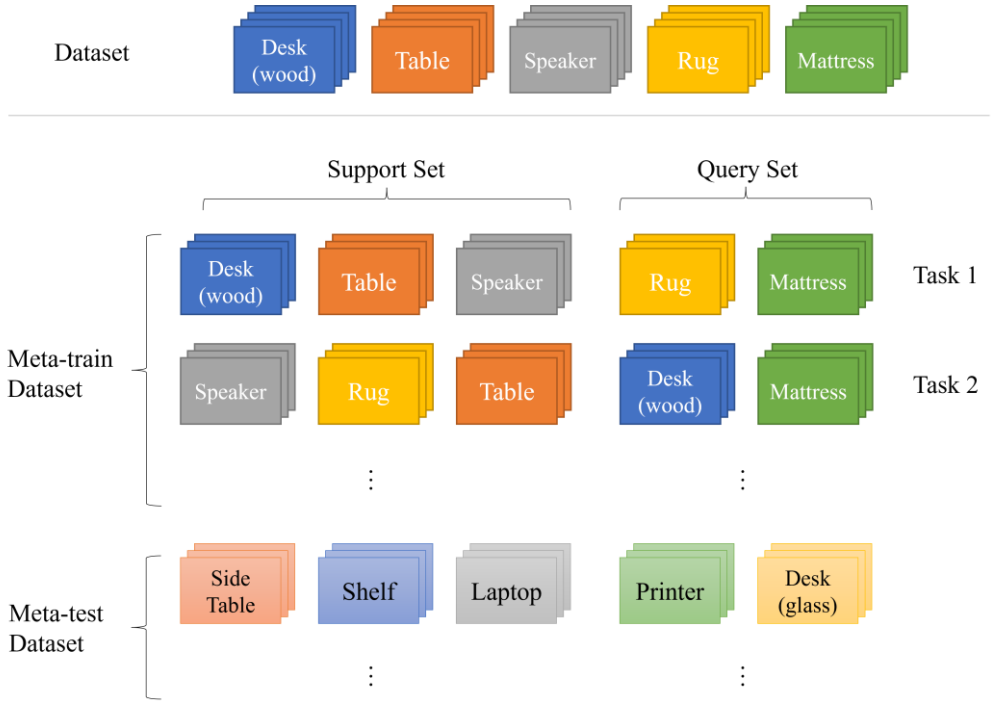


Figure 1. Example of 3-way 3-shot Few-Shot learning task.

4. Few-Shot Learning-Based Material Recognition

This section introduces FSMR, a few-shot learning-based material recognition method using smartphones. The existing research on deep learning-based contact material recognition using smartphones faces challenges as it requires large amounts of training data to classify new materials not previously trained in the model. FSMR addresses the difficulty of data collection by employing data augmentation techniques and few-shot learning. Data augmentation increases the amount of data available for training few-shot learning models, preventing overfitting and significantly reducing the data needed to train on new materials. The overall process of the proposed FSMR is illustrated in Figure 2, consisting of data collection, data augmentation, and model training. This section provides a detailed explanation of each component.

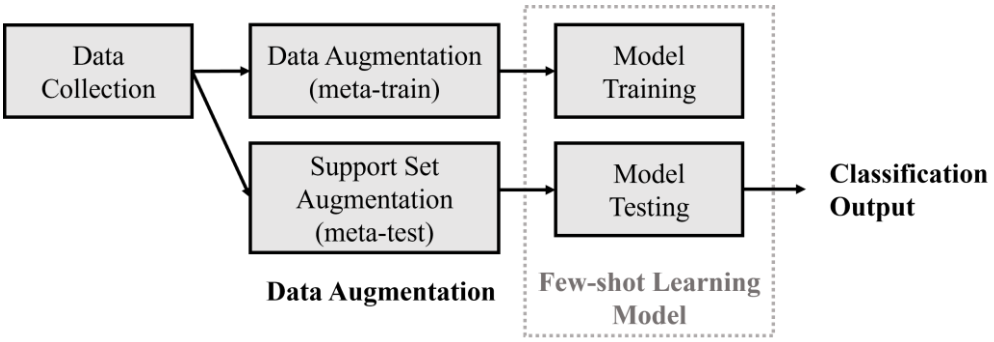


Figure 2. Few-Shot learning-based material recognition overview.

4.1. Data Collection

This section describes the method for collecting the data required to train the few-shot learning model for material recognition. When a smartphone is in contact with an object, the vibration signal generated by the smartphone is altered due to the attenuation, reflection, and diffraction caused by the object [29], as shown in Figure 3. Z_1 and Z_2 represent the characteristic impedances of a smartphone and a desk, respectively. $f_1(t)$ is

the smartphone's vibration signal, $f_2(t)$ is the signal diffracted upon reaching the desk, and $g_1(t)$ is the signal reflected to the smartphone from the desk. The diffracted signal $f_2(t)$ and the reflected signal $g_1(t)$ can be represented mathematically as follows:

$$f_2(t) = \frac{2Z_1}{Z_1 + Z_2} f_1(t) \quad (1)$$

$$g_1(t) = \frac{Z_1 - Z_2}{Z_1 + Z_2} f_1(t) \quad (2)$$

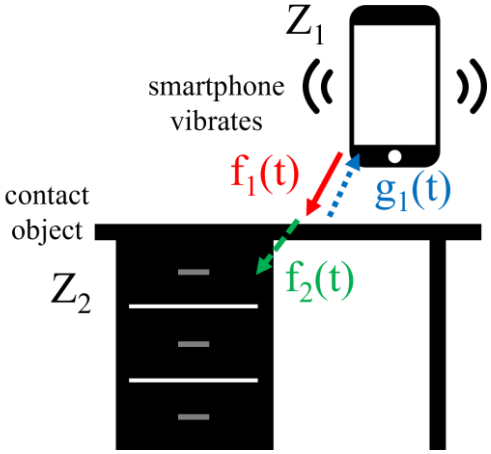


Figure 3. Representation of smartphone's vibration signals during interaction with material.

The attenuation, reflection, and diffraction of vibrations transmitted to the smartphone vary depending on the unique characteristic impedance of the material. Based on this principle, FSMR learns to classify materials by analyzing accelerometer data collected during vibration events. The data collection process is as follows: (1) Generate Vibrations: The smartphone generates vibrations for one second. (2) Collect Accelerometer Data: Simultaneously, the accelerometer collects data. This process is repeated by placing the smartphone on various objects to collect several seconds of data for each material. The collected data then undergo data augmentation and few-shot learning processes to classify the materials in contact with the smartphone.

4.2. Data Augmentation

Deep learning models typically require large amounts of data for effective training. However, the process of collecting real-world data is time-consuming and expensive, posing practical challenges. Additionally, training a deep learning model with a small dataset can lead to overfitting, where the model performs well on training data but poorly generalizes to new data, ultimately reducing its overall performance. Therefore, additional data augmentation techniques are necessary to address the challenges of limited data and enhance usability for real-world applications. Data augmentation is a technique that generates new data by applying various transformations to the existing data while keeping the original labels. This study utilizes data augmentation to prevent overfitting during the training of few-shot learning models and to achieve high accuracy during testing.

According to Equations (1) and (2), the unique characteristics of the contacting material allow the material to be classified by using reflected vibration signals. Observation of the collected accelerometer data revealed that when the smartphone was in contact with the same material, the vibration exhibited similar waveforms with slight variations in amplitude. Furthermore, the waveforms varied depending on whether the material vibrated along with the smartphone (e.g., desks or drawers) or remained stationary (e.g.,

sofas or blankets). Materials prone to vibrations showed changes in the waveform when the smartphone's position was altered, whereas materials less prone to vibrations exhibited minimal changes in the waveform regardless of position. In other words, for any material, vibration data collected in contact with the same material typically exhibit similar waveforms but different amplitudes. This suggests that slightly altering the amplitude can serve as a data augmentation method applicable across different materials. Also, to examine the changes in accelerometer data caused by variations in temperature and humidity around the material, we conducted the following experiments. Accelerometer data were collected by vibrating a smartphone placed on various objects, including a Bluetooth speaker, a mouse pad, and a wooden desk. During data collection, various environmental conditions were created, such as heating or cooling the objects or increasing humidity by spraying water with a mist sprayer. The detailed experimental setup and results are presented in Table 1. After measuring and calculating metrics such as the mean, maximum, minimum, and standard deviation under different conditions, the results showed that the effects of temperature and humidity on the accelerometer data were negligible.

Table 1. Analysis of reflected vibrations in materials using accelerometer data under various environments.

Material	Indicator	Accelerometer	Temperature (40%)			Humidity (23°)	
			2°	23°	50°	40%	80%
Bluetooth Speaker	Average	*X*-axis	−0.008882	0.000184	−0.001446	−0.008727	0.000613
		Y-axis	0.004422	0.001439	0.004802	0.001380	0.004385
		Z-axis	0.000602	0.000845	0.002273	−0.000859	0.000838
	Max	*X*-axis	0.025790	0.059895	0.036289	0.085450	0.028937
		Y-axis	0.048728	0.034102	0.040494	0.051139	0.047338
		Z-axis	0.113164	0.151749	0.135862	0.180153	0.119132
	Min	*X*-axis	−0.041900	−0.057658	−0.038760	−0.095913	−0.034055
		Y-axis	−0.034162	−0.032984	−0.031085	−0.029767	−0.038391
		Z-axis	−0.103663	−0.143733	−0.134073	−0.153780	−0.116957
	Standard Deviation	*X*-axis	0.017023	0.030261	0.018867	0.055076	0.013139
		Y-axis	0.018958	0.016441	0.016765	0.017406	0.022619
		Z-axis	0.065961	0.082885	0.081750	0.090561	0.068292
Mousepad	Average	*X*-axis	0.002603	0.001329	0.003758	0.000126	0.003021
		Y-axis	−0.005401	−0.004400	−0.014829	−0.005039	−0.003797
		Z-axis	−0.000120	0.001314	0.000703	0.001638	0.000269
	Max	*X*-axis	0.108383	0.091903	0.095357	0.088185	0.094049
		Y-axis	0.080062	0.076471	0.056788	0.088366	0.066936
		Z-axis	0.142771	0.145910	0.147636	0.170978	0.106804
	Min	*X*-axis	−0.115528	−0.087210	−0.084311	−0.080832	−0.084881
		Y-axis	−0.096576	−0.081857	−0.095087	−0.091243	−0.061099
		Z-axis	−0.140719	−0.146928	−0.124035	−0.168050	−0.108000
	Standard Deviation	*X*-axis	0.068634	0.060658	0.059236	0.055612	0.058391
		Y-axis	0.051577	0.047986	0.040319	0.053565	0.032382
		Z-axis	0.096678	0.102472	0.085793	0.116832	0.073531
Wooden Desk	Average	*X*-axis	−0.010640	−0.003851	−0.011936	−0.004946	−0.003851
		Y-axis	0.015642	−0.014120	−0.019988	−0.012705	−0.014121
		Z-axis	−0.000763	−0.000397	0.001002	0.000086	−0.000397
	Max	*X*-axis	0.102039	0.093620	0.087946	0.087561	0.093621
		Y-axis	0.086843	0.018715	0.054478	0.013586	0.018716
		Z-axis	0.082414	0.085871	0.160721	0.086502	0.085872
	Min	*X*-axis	−0.127496	−0.095567	−0.357540	−0.099718	−0.095568
		Y-axis	−0.033584	−0.043840	−1.073221	−0.042039	−0.043841
		Z-axis	−0.076922	−0.094424	−0.098207	−0.080541	−0.094424
	Standard Deviation	*X*-axis	0.067431	0.053306	0.070767	0.055658	0.053306
		Y-axis	0.023407	0.015173	0.146325	0.013029	0.015174
		Z-axis	0.045157	0.050125	0.053638	0.050328	0.050126

Figure 4 illustrates portions of vibration data collected when the smartphone's position was changed on the same object. While the overall waveforms remained largely unchanged, slight variations in amplitude were observed. Generally, the amplitude of the *Y*-axis was the smallest, while the *Z*-axis amplitude was the largest. The *Z*-axis amplitude showed relatively consistent behavior, whereas the *X*-axis amplitude exhibited more significant variations in some cases. Based on these observations, data augmentation was implemented through scaling, where the amplitudes of the *X*-, *Y*-, and *Z*-axes were adjusted within random ranges as follows. For the *X*-axis, which exhibited larger amplitude variations with position changes, the amplitude was scaled by a factor randomly chosen between 0.5 and 1.5. For the *Y*- and *Z*-axes, which showed smaller variations, the amplitude was scaled by a factor randomly chosen between 0.8 and 1.2. All adjustments were made with uniform probabilities within the specified ranges. The results of the augmented data after applying scaling are shown in Figure 5.

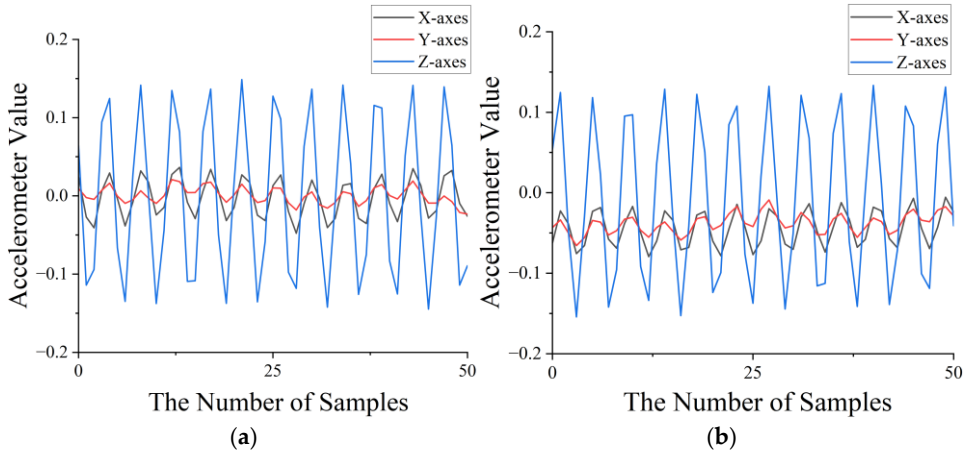


Figure 4. Accelerometer data collected with the smartphone placed on a blanket. (**a**) Data collected on the blanket. (**b**) Data collected at a different position on the same blanket.

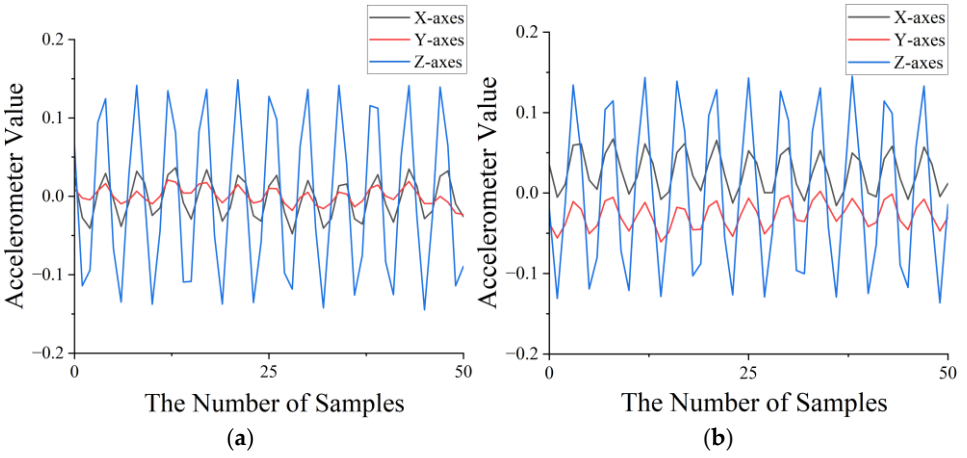


Figure 5. Accelerometer data collected with a smartphone placed on a blanket. (**a**) Original data. (**b**) Augmented data by applying scaling to the original data.

In the FSMR system, data augmentation plays a crucial role in enhancing the performance and practicality of the few-shot learning model. The process is based on the observation that vibration signals generated when a smartphone contacts the same material exhibit similar waveforms with slight variations in amplitude. This behavior is influenced by whether the material vibrates along with the smartphone (e.g., desks or drawers) or remains stationary (e.g., sofas or blankets), as well as the position of the smartphone on the

material. To leverage these characteristics, data augmentation is performed by adjusting the amplitude of the vibration signals collected through the smartphone's accelerometer. The amplitude scaling process is mathematically defined as follows:

$$A'_X = A_X \cdot r_X, \quad A'_Y = A_Y \cdot r_Y, \quad A'_Z = A_Z \cdot r_Z \tag{3}$$

where A_X, A_Y, A_Z represent the original amplitudes of the vibration signals, and r_X, r_Y, r_Z are scaling factors randomly sampled from the following ranges:

$$r_X \sim U(0.5, 1.5), \quad r_Y \sim U(0.8, 1.2), \quad r_Z \sim U(0.8, 1.2) \tag{4}$$

As a result, A'_X, A'_Y, A'_Z are the augmented amplitudes.

This data augmentation process is utilized during both the training and testing phases to enhance the few-shot learning model's performance. During training, it prevents overfitting by generating diverse tasks using both the raw data collected via smartphones and the augmented data. This process improves the model's ability to generalize effectively. In the testing phase, data augmentation is applied to the support set of test tasks to increase the available data, enabling the model to classify new materials more accurately with minimal input. This data augmentation allows users to collect significantly less data for new materials compared to the large datasets required by conventional deep learning models. By providing a more diverse and representative dataset, this approach ensures the model can adapt quickly and maintain high classification accuracy, making FSMR highly practical for real-world scenarios.

4.3. Few-Shot Learning for Material Recognition

The FSMR few-shot learning model proposed in this study is designed based on ProtoNet [28]. The model's structure and hidden layers are illustrated in Figures 6 and 7. The convolutional block consists of a conv2D layer, batch normalization, and a ReLU activation layer. The conv2D layer is configured with 100 channels and a kernel size of three. The embedding network is constructed with four convolutional blocks, along with max-pooling and dropout layers. In training, training tasks augmented by scaling pass through the embedding network to form class prototypes via the clustering of the support set. These class prototypes establish the embedding space, which is then used for classifying the query set. The Euclidean distance between the class prototypes and query points is used to compute the loss. The model updates its weights in a direction that minimizes the loss. During evaluation, the trained embedding network is used. Like training, the support set of evaluation tasks forms the embedding space, and the query set is classified within this space to assess performance.

The FSMR system integrates a few-shot learning model to enable material classification with minimal data. Few-shot learning in FSMR employs a task-based meta-learning framework, where each task τ consists of a support set S and a query set Q:

$$\tau = \{S, Q\} \tag{5}$$

$$S = \{(x_i, y_i)\}_{i=1}^{K} \tag{6}$$

$$Q = \{(x_j, y_j)\}_{j=1}^{M} \tag{7}$$

Here, S contains K labeld samples used for model adaptation, and Q contains M unlabeled samples used for evaluation. The purpose of few-shot learning is to optimize the

model f_θ to minimize classification errors on Q after adapting to S. The meta-learning loss is defined as

$$\mathcal{L}_{meta} = \mathbb{E}_\tau \left[\mathcal{L}_{query}(f_{\theta^*}(S), Q) \right] \tag{8}$$

where $\mathcal{L}_{query}$ denotes the classification loss on the query set, and θ^* represents the model parameters adapted after training on S. The detailed procedure of FSMR is presented in Algorithm 1.

In FSMR, meta-learning comprises training (meta-train) and testing (meta-test) phases. During the training phase, diverse training tasks are generated using the augmented training data, improving the model's generalization capability. During the testing phase, augmented support sets are generated for new materials, allowing the model to adapt quickly and achieve optimal classification performance. This approach enables the efficient use of data, achieving high accuracy with significantly less data compared to conventional deep learning models. The integration of few-shot learning into FSMR enhances its practicality and scalability, enabling effective material classification with minimal data, while allowing rapid adaptation to new materials.

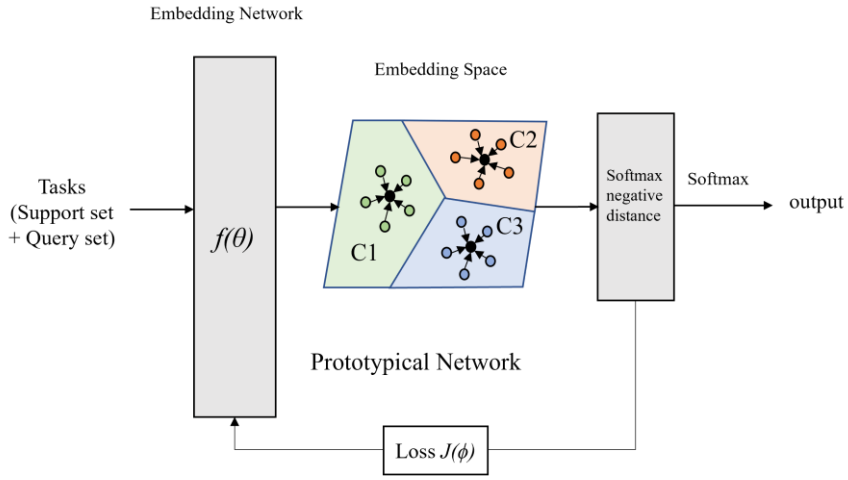


Figure 6. Structure of ProtoNet-based Few-Shot learning model.

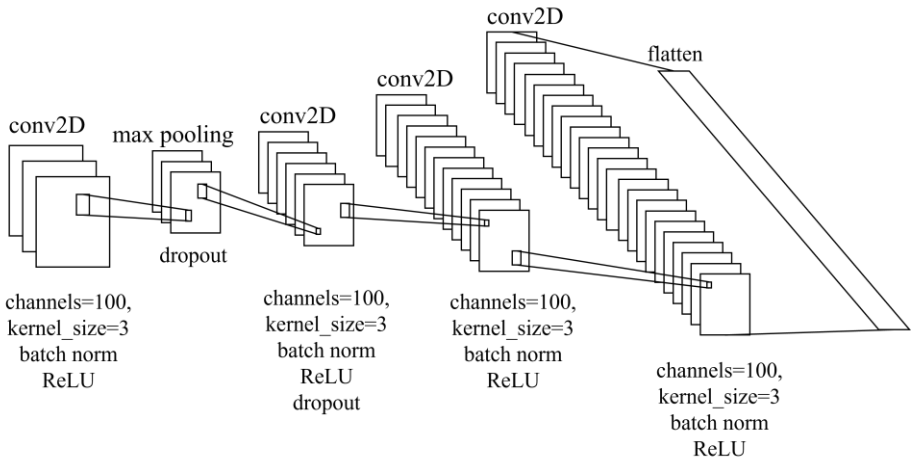


Figure 7. Structure of embedding network.

Algorithm 1 Few-Shot Learning in FSMR

Require: Training data D_{train}, Test data D_{test}, Few-shot learning model f_θ with initial parameters θ, Number of training tasks N_{train}, Number of test tasks N_{test}
Ensure: Updated model parameters θ, Classification results for test tasks
1: **Meta-training Phase:**
2: **for** $i = 1$ to N_{train} **do**
3: Sample a training task $\tau_i = \{S,\ Q\}$ from D_{train}
4: S: Support set (labeled samples)
5: Q: Query set (unlabeled samples)
6: Compute task-specific parameters θ_i by optimizing on S:

$$\theta_i = \theta - \alpha \nabla_\theta \mathcal{L}(S, \theta)$$

7: Compute query loss $\mathcal{L}(Q, \theta_i)$ on Q
8: **end for**
9: Update meta-parameters θ by minimizing the meta-loss:

$$\mathcal{L}_{meta} = \frac{1}{N_{train}} \sum_{i=1}^{N_{train}} \mathcal{L}(Q, \theta_i)$$

10: **Meta-testing phase:**
11: **for** $j = 1$ to N_{test} **do**
12: Sample a test task $\tau_i = \{S_{test}, Q_{test}\}$ from D_{test}
13: Apply data augmentation to S_{test} to generate augmented support set S_{aug}
14: Adapt model parameters θ_j using S_{aug}

$$\theta_j = \theta - \alpha \nabla_\theta \mathcal{L}(S_{aug}, \theta)$$

15: Classify query set Q_{test} using the adapted parameters θ_j
16: **end for**
17: **return** Updated model parameters θ, Classification results for τ_{test}

5. Performance Evaluation

This section introduces the experiments conducted to evaluate the performance of the proposed FSMR. Section 5.1 describes the data collection methods and how the collected data were input into the model. Section 5.2 presents the experimental results to assess the performance of FSMR. The effects of the few-shot learning model and data augmentation are evaluated, and the results of FSMR are compared with those of existing studies.

5.1. Data Collection for Performance Evaluation

The smartphone used for data collection was a Samsung Galaxy S20. An application was developed using Android Studio to generate vibrations and collect accelerometer data. The smartphone generated vibrations for one second, during which accelerometer data were collected. A total of 190 training samples were collected for each of the 55 materials; evaluation samples were collected for each material. For each contacting object, the surface was divided into several sections, and multiple samples were collected from each section. The smartphone's position was changed randomly during the data collection process to enhance variability. Table 2 summarizes the collected data. The table includes the type (distinguishing training and evaluation data), N (the number of collected classes), T (the collection time in seconds), and Class (the objects in contact with the smartphone).

Table 2. Overview of the dataset collected for performance evaluation.

Dataset		
Type	**Train**	**Test**
N	45	10
T	190	190
Class	bag1(leather1), bag2(leather2), cd player, chair1, clipboard, computer, cushion, desk1, desk2, desk3(glass1), desk4(glass2), desk5, deskpad1, deskpad2, desktop, doormat, drawer1, drawer2, drawingtablet, hand(sitting), hand(walking), hand1, laptop1, ledge1(concrete), mattress1, mattress2, mattress3, mattresspad, monitorstand1, monitorstand2, mousepad1, mousepad2, mousepad3, newspaper, pillow1, pillow2, pocket1, pocket2(jacket), sofa1, speaker, table1, table2(small), table3, table4, trivet(rubber)	blanket, bluetooth speaker, chair2, desk6(wood), drawer3, laptop2, ledge2, pillow3, sofa2, sofa3

5.2. Performance Evaluation of FSMR

In Section 5.2.1, improvements achieved solely by the model without applying data augmentation are evaluated. In Section 5.2.2, the performance of data augmentation is assessed. In Section 5.2.3, the performance of FSMR is compared with VibePhone [6], an existing study that uses vibrations and accelerometers to recognize objects in contact with a smartphone. The comparison focuses on the amount of data, emphasizing the reduction in training data while maintaining performance.

5.2.1. Few-Shot Learning Evaluation

Figure 8 illustrates the classification performance of CNN and few-shot learning models in a graphical format. This experiment was conducted to evaluate whether the few-shot learning model can classify effectively with small datasets, comparing its performance to that of the CNN model, which is commonly used for classification tasks. The total number of classes for classification was 10, and the performance was evaluated under two conditions: when trained with a small dataset of 15 s per class and when trained with a larger dataset of 70 s per class.

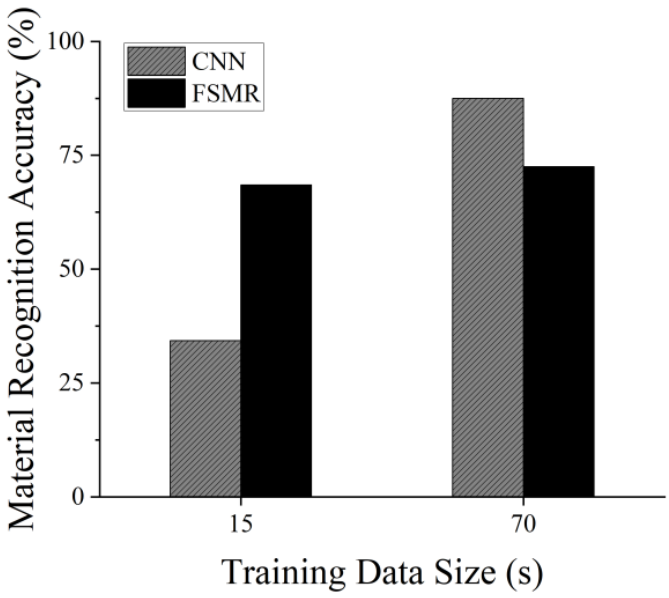


Figure 8. Material recognition accuracy of CNN and FSMR trained by 2 datasets that have different sizes.

When trained with the larger dataset (70 s per class), the CNN model achieved 87.5% accuracy, outperforming the few-shot learning model, which achieved 72.5%. However, the practical application of the CNN model in real context-aware services remains limited due to its data requirements. Conversely, when trained with the smaller dataset (15 s per class), the few-shot learning model achieved a significantly higher accuracy of 68.5% compared to the CNN model's 34.29%. This result demonstrates that a dataset of 15 s per class is insufficient for conventional deep learning models to achieve reasonable classification accuracy. However, the few-shot learning model can achieve a certain level of classification accuracy even with minimal data, thus validating the effectiveness of the few-shot learning approach.

Figure 9 presents the results of applying various training configurations to the few-shot learning model. This experiment aimed to identify the optimal N-way K-shot configuration for classification purposes. The number of classes ranged from a minimum of 3-way to a maximum of 10-way, and the number of data points used for training ranged from a minimum of 1-shot to a maximum of 15-shot. For N-way training, N random classes were selected from a total of 10 classes, and the classification accuracy for these selected classes was evaluated. This process was repeated 1000 times with random class selections, and the average accuracy was calculated to evaluate the performance of N-way classification across various class combinations.

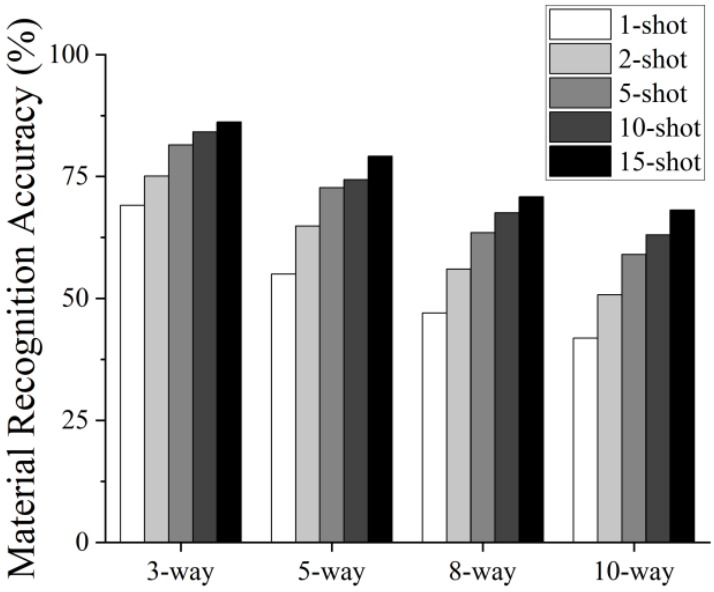


Figure 9. Material recognition accuracy of FSMR using different N-way K-shot configurations.

The results show that the fewer the number of classes to classify and the greater the number of data points used for training, the higher the classification accuracy. The best-performing configurations were 3-way 10-shot and 3-way 15-shot, achieving accuracies of 86.12% and 86.17%, respectively. These results align with the general pattern observed in classification models, where smaller class numbers and larger datasets lead to higher accuracy. However, it is important to note that the training data used in this experiment were only 15 s per class, and the accuracy difference between 10-shot and 15-shot was minimal. This highlights the data efficiency of few-shot learning. The worst-performing configuration was 10-way 1-shot, where the number of classes was high, and the amount of training data was minimal, resulting in an accuracy of 40.13%. In contrast, with more data in the 10-way 15-shot configuration, the model achieved a decent classification accuracy of 68.15%. A particularly noteworthy result was observed for the five-way configuration.

Despite the moderate number of classes to classify, the average classification accuracy for all combinations of 10-shot and 15-shot exceeded 80%. This indicates that a five-way configuration is well suited for setting up the training dataset in few-shot learning.

In the previous experiment, the maximum number of evaluation classes was 10. For the 3-way, 5-way, and 8-way configurations, experiments were conducted by randomly selecting 3, 5, and 8 classes from the 10 total classes, respectively. The results showed variations in accuracy depending on the specific combinations of randomly selected classes. Table 3 presents the classification accuracy achieved with different class combinations for the 5-way 10-shot configuration. Among the 252 possible combinations of selecting 5 classes out of 10, the best-performing combination achieved a high accuracy of 93.49%, while the worst-performing combination achieved a low accuracy of 61.07%. Objects such as Bluetooth speakers showed a lower accuracy in combinations that included them, as their small surface area caused significant waveform variations when the smartphone was rotated in place. Additionally, objects with soft, stable surfaces (e.g., blankets, sofas) and objects with hard, unstable surfaces (e.g., metal drawers) exhibited distinct waveform patterns. Combinations of objects with different material properties generally resulted in higher accuracy. Developing methods to minimize the performance differences caused by these class combinations could further improve the overall classification accuracy.

Table 3. Material recognition accuracy of top 3 and bottom 3 combinations of classes when 5-way 10-shot learning.

Rank	Class	Accuracy (%)
1	blanket, drawer3, labtop2, ledge2, sofa2	93.49
2	blanket, drawer3, labtop2, ledge2, sofa3	92.15
3	desk(wood), blanket, drawer3, labtop2, ledge2, sofa2	91.28
250	bluetooth speaker, desk(wood), drawer3, ledge2, sofa3	62.29
251	bluetooth speaker, chair2, desk(wood), drawer3, ledge2	62.09
252	bluetooth speaker, chair2, desk(wood), drawer3, pillow3	61.07

5.2.2. Data Augmentation Evaluation

Figure 10 shows the results of applying the proposed data augmentation technique to expand the meta-training data and evaluation support sets. Evaluations were conducted for 3-way, 5-way, 8-way, and 10-way configurations. The size of the evaluation support sets was increased by integer multiples: one-shot data were tripled, two-shot data were quintupled, and five-shot data were doubled. The results, both before and after augmentation, are presented. The results indicate that tripling the 1-shot data improved accuracy by an average of 4.05% across the 3- to 10-way configurations. Quintupled two-shot data achieved an average accuracy improvement of 3.61%, and doubling the five-shot data improved accuracy by 2.16%. The highest accuracy was achieved with the 3-way configuration when the support set was increased from 5-shot to 10-shot, achieving an accuracy of 83.63%. This experiment demonstrates that scaling through data augmentation effectively improves classification performance. By experimenting with various augmentation scales, the results suggest that the proposed method is a highly suitable improvement for implementing real-world applications for users.

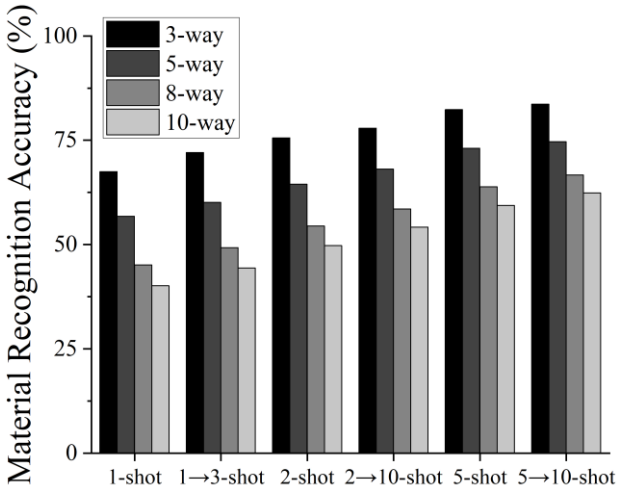


Figure 10. Material recognition accuracy of FSMR when augmenting the evaluation support set.

5.2.3. Comparison with Previous Research

Table 4 compares the performance of FSMR, proposed in this paper, with that of VibePhone, a study that classifies objects in contact with a smartphone using vibrations and accelerometers. Both models were evaluated using six classes. VibePhone trained on the entirety of data collected over 180 s, whereas FSMR was trained only on datasets of 80 s or less and used a tripled evaluation support set for testing.

Table 4. Material recognition accuracy comparison with VibePhone.

	VibePhone [6]	**FSMR**					
Training data size (s)	180	2	5	10	30	50	80
Accuracy (%)	85	71.94	76.34	83.01	90.73	91.79	91.91

The results show that the VibePhone achieved an average classification accuracy of 85%, demonstrating high performance. However, it is important to note that the training dataset size was significantly large at 180 s. FSMR, when trained on data collected over 2 s, achieved a 71.94% accuracy, which appears insufficient for practical use. However, with 5 s of data, FSMR achieved 76.34% accuracy, and with 10 s of data, it reached 83.01%. The training durations of 5 and 10 s are reasonably short for practical user registration, and the nearly 80% accuracy achieved with 10 s makes FSMR suitable for real-world applications. For longer training durations of 30, 50, and 80 s (although less practical for users) the results can serve as benchmarks for comparison with previous studies. FSMR achieved 90.73%, 91.79%, and 91.91% accuracy for these durations, respectively. These figures are comparable to the VibePhone's performance and demonstrate that FSMR significantly reduces the amount of data required for training, while maintaining a competitive accuracy. Although FSMR did not outperform the VibePhone in any scenario, the ability to achieve a similar accuracy with substantially less training data highlights its potential as a more practical solution for real-world applications.

Table 5 presents the results of response time measurements for VibePhone and FSMR. The same models and training datasets as in the previous experiments were used, with the input data size for VibePhone set to 1 s. The results show that VibePhone achieved a response time of approximately 19 ms for 1 s input data, which is faster than FSMR's response times of 22 ms and 26 ms for input sizes of 2 s and 5 s, respectively. However, even for larger input sizes, FSMR maintained a response time within 80 ms for 50 s input data and 115 ms for 80 s input data, a difference of only 30 ms. This response time is

imperceptibly short for human users. Considering that FSMR achieved over 90% accuracy in the previous experiment with a training dataset of only 30 s, these results demonstrate the exceptional performance of FSMR.

Table 5. Response time comparison with VibePhone.

	VibePhone [6]	FSMR					
Test data size (s)	1	2	5	10	30	50	80
Inference time (ms)	19	22	26	31	53	74	115

6. Conclusions

This study aims to develop a system that leverages smartphone vibrations and accelerometers to identify materials in contact with the device, enabling object classification for use in context-aware services. The existing research, VibePhone, which recognizes objects using vibrations and accelerometers, requires 180 s of training data for deep learning. In contrast, the proposed FSMR achieves a similar performance with much shorter data collection durations. Reducing the data collection time allows users to directly register contexts in real-world applications, facilitating personalized context-aware services. The final evaluation results show that FSMR achieved classification accuracies of 71.94%, 76.34%, 83.01%, 90.73%, 91.79%, and 91.91% when trained on data collected over 2, 5, 10, 30, 50, and 80 s, respectively, for six objects. In Section 5.2, the few-shot learning results indicate that there was a minimal performance difference between 10-shot and 15-shot configurations. This suggests diminishing returns when using more than 10 data points per class. For practical applications, collecting data for 10 s or less appears to be appropriate. Moreover, the performance for 10-way classification was relatively poor, indicating that limiting the number of classes in real-world applications may be beneficial. Additionally, performance varies depending on class combinations. When implementing an actual application, improvements to the system may be required to minimize the impact of such variations on classification accuracy. In the future, we plan to develop context-aware services utilizing location information determined by FSMR. These context-aware services are expected to provide a wider variety of services for users indoors, as more accurate location information can be utilized for the understanding of context. Additionally, by reducing the amount of data users need to collect, practicality has been enhanced, making the actual implementation of FSMR-based context-aware services feasible.

Author Contributions: Conceptualization, S.K.; formal analysis, S.K.; funding acquisition, S.K.; methodology, J.Y., Y.K. and S.K.; project administration, S.K.; software, J.Y. and Y.K.; supervision, S.K.; validation, J.Y. and Y.K.; visualization, J.Y.; writing—original draft, J.Y. and Y.K.; and writing—review and editing, S.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partly supported by the Innovative Human Resource Development for Local Intellectualization program through an Institute of Information and Communications Technology Planning and Evaluation (IITP) grant funded by the Korean government (MSIT) (IITP-2025-2020-0-01462, 50%). This work was supported by a National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. 2022R1A5A8026986, 40%), and this paper was supported by a Korea Institute for Advancement of Technology (KIAT) grant funded by the Korean Government (MOTIE) (P0020536, HRD Program for Industrial Innovation, 10%).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available in the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Casillo, M.; Colace, F.; Conte, D.; Lombardi, M.; Santaniello, D.; Valentino, C. Context-aware recommender systems and cultural heritage: A survey. *J. Ambient. Intell. Humaniz. Comput.* **2023**, *14*, 3109–3127. [CrossRef]
2. Jain, A.; Bhutani, P. A Study of Context-aware Systems. In Proceedings of the International Conference on Artificial Intelligence and Networking, New Delhi, India, 7 November 2024. [CrossRef]
3. Yang, J.; Munguia-Tapia, E.; Gibbs, S. Efficient in-pocket detection with mobile phones. In Proceedings of the 2013 ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication, Zurich, Switzerland, 8–12 September 2013; pp. 31–34. [CrossRef]
4. Yeo, H.; Lee, J.; Bianchi, A.; Harris-Birtill, D.; Quigley, A. SpeCam: Sensing surface color and material with the front-facing camera of a mobile device. In Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services, Vienna, Austria, 4–7 September 2017; pp. 1–9. [CrossRef]
5. Hwang, S.; Wohn, K. VibroTactor: Low-cost placement-aware technique using vibration echoes on mobile devices. In Proceedings of the Companion Publication of the 2013 International Conference on Intelligent User Interfaces Companion, Santa Monica, CA, USA, 19–22 March 2013; pp. 73–74. [CrossRef]
6. Cho, J.; Hwang, I.; Oh, S. Vibration-based surface recognition for smartphones. In Proceedings of the 2012 IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, Seoul, Republic of Korea, 19–22 August 2012; pp. 459–464. [CrossRef]
7. El Khaddar, M.A.; Boulmalf, M. Smartphone: The ultimate IoT and IoE device. In *Smartphones from an Applied Research Perspective*; IntechOpen: London, UK, 2017; pp. 137–160.
8. Sarker, I.H. Mobile data science: Towards understanding data-driven intelligent mobile applications. *EAI Endorsed Trans. Scalable Inf. Syst.* **2018**, *18*, e1. [CrossRef]
9. Schilit, B.N.; Theimer, M.M. Disseminating active map information to mobile hosts. *IEEE Netw.* **1994**, *8*, 22–32. [CrossRef]
10. Dey, A.K. Understanding and using context. *Pers. Ubiquit. Comput.* **2001**, *5*, 4–7. [CrossRef]
11. Sarker, I.H. Context-aware rule learning from smartphone data: Survey, challenges and future directions. *J. Big Data* **2019**, *6*, 95. [CrossRef]
12. Sarker, I.H. BehavMiner: Mining user behaviors from mobile phone data for personalized services. In Proceedings of the 2018 IEEE International Conference on Pervasive Computing and Communications (PerCom 2018), Athens, Greece, 19–23 March 2018.
13. Bell, S.; McDiarmid, A.; Irvine, J. Nodobo: Mobile phone as a software sensor for social network research. In Proceedings of the 2011 IEEE 73rd Vehicular Technology Conference (VTC Spring), Budapest, Hungary, 15–18 May 2011; pp. 1–5. [CrossRef]
14. Pielot, M. Large-scale evaluation of call-availability prediction. In Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '14), Seattle, WA, USA, 13–17 September 2014; pp. 933–937. [CrossRef]
15. Eagle, N.; Pentland, A. Reality mining: Sensing complex social systems. *Pers. Ubiquit. Comput.* **2006**, *10*, 255–268. [CrossRef]
16. Sarker, I.H.; Colman, A.; Kabir, M.A.; Han, J. Behavior-oriented time segmentation for mining individualized rules of mobile phone users. In Proceedings of the 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA), Montreal, QC, Canada, 17–19 October 2016; pp. 488–497. [CrossRef]
17. Sarker, I.H.; Kabir, M.A.; Colman, A.; Han, J. Designing architecture of a rule-based system for managing phone call interruptions. In Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers (UbiComp'17), Maui, HI, USA, 11–15 September 2017; pp. 898–903. [CrossRef]
18. Zhu, H.; Chen, E.; Xiong, H.; Yu, K.; Cao, H.; Tian, J. Mining mobile user preferences for personalized context-aware recommendation. *ACM Trans. Intell. Syst. Technol.* **2014**, *5*, 4. [CrossRef]
19. Xu, X.; Yu, J.; Chen, Y.; Hua, Q.; Zhu, Y.; Chen, Y.; Li, M. TouchPass: Towards behavior-irrelevant on-touch user authentication on smartphones leveraging vibrations. In Proceedings of the 26th Annual International Conference on Mobile Computing and Networking (MobiCom '20), London, UK, 21–25 September 2020; pp. 1–13. [CrossRef]
20. Lu, J.; Wu, D.; Mao, M.; Wang, W.; Zhang, G. Recommender system application developments: A survey. *Decis. Support Syst.* **2015**, *74*, 12–32. [CrossRef]
21. Um, T.T.; Pfister, F.M.J.; Pichler, D.; Endo, S.; Lang, M.; Hirche, S.; Fietzek, U.; Kulić, D. Data augmentation of wearable sensor data for Parkinson's disease monitoring using convolutional neural networks. In Proceedings of the 19th ACM International Conference on Multimodal Interaction (ICMI '17), Glasgow, UK, 13–17 November 2017; pp. 216–220.
22. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [CrossRef]
23. Lake, B.; Salakhutdinov, R.; Gross, J.; Tenenbaum, J. One-shot learning of simple visual concepts. In Proceedings of the 33rd Annual Meeting of the Cognitive Science Society (CogSci'11), Boston, MA, USA, 20–23 July 2011. Available online: https://escholarship.org/uc/item/4ht821jx#author (accessed on 26 December 2024).

24. Koch, G.; Zemel, R.; Salakhutdinov, R. Siamese neural networks for one-shot image recognition. In Proceedings of the ICML Deep Learning Workshop, Lille, France, 6–11 July 2015.
25. Miller, E.G.; Matsakis, N.E.; Viola, P.A. Learning from one example through shared densities on transforms. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2000), Hilton Head Island, SC, USA, 13–15 June 2000; pp. 464–471. [CrossRef]
26. Ravi, S.; Larochelle, H. Optimization as a model for few-shot learning. In Proceedings of the International Conference on Learning Representations (ICLR 2016), San Juan, Puerto Rico, 2–4 May 2016.
27. Vinyals, O.; Blundell, C.; Lillicrap, T.; Kavukcuoglu, K.; Wierstra, D. Matching networks for one-shot learning. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 3630–3638.
28. Snell, J.; Swersky, K.; Zemel, R.S. Prototypical networks for few-shot learning. *arXiv* **2017**, arXiv:1703.05175.
29. Sarker, I.H.; Abushark, Y.B.; Khan, A.I.; Colman, A. Mobile deep learning: Exploring deep neural network for predicting context-aware smartphone usage. *SN Comput. Sci.* **2021**, *2*, 146. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.